



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA EN INFORMÁTICA

EVALSIM: SISTEMA PARA LA EXTRACCIÓN DE INDICADORES DE INTERACCIÓN EN MUNDOS VIRTUALES

Raúl Gómez Sánchez

10 de septiembre de 2017



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO EN INFORMÁTICA

EVALSIM: SISTEMA PARA LA EXTRACCIÓN DE INDICADORES DE INTERACCIÓN EN MUNDOS VIRTUALES

- Departamento: Ingeniería Informática
- Directores del proyecto: Antonio Balderas Alberico y Manuel Palomo Duarte
- Autor del proyecto: Raúl Gómez Sánchez

Cádiz, 10 de septiembre de 2017

Fdo: Raúl Gómez Sánchez

Agradecimientos

Me gustaría dedicar este texto a mi familia y en especial a mis padres, José Gómez y Beatriz Sánchez, que han hecho muchos sacrificios para que pudiese llegar este momento.

También debo agradecer este proyecto a los profesores Anke Berns, Manuel Palomo y Antonio Balderas; que han sido los que me han ofrecido esta oportunidad.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2017 Raúl Gómez Sánchez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:
emacs.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:
quicksort.

Cuando nos refiramos a un bloque de código, utilizaremos la notación:

```
1  //This is a comment
2  for (String ent:entities) {
3      var targets_list = new ArrayList()
4      var target_step = "TARGET"
5      switch (ent) {
6          case Activity.MEMORY_SATURN_LITERAL.literal: {
7              if (e.entities.length != 0)
8                  throw new IllegalArgumentException('Not valid Entity for this type: ' +
9                      ent)
10             }
11             case Activity.HIDDEN_ROOM_LITERAL.literal:{
12                 steps.add(new Step('TableInput', 'N', 1, x2, y2, ent, e))
13                 steps.add(new Step('Calculator', 'N', 1, x2, y2, ent, e))
14                 hops.add(new
15                     Hop(steps.get(steps.size-2).name, steps.get(steps.size-1).name, 'Y'))
```


Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Descripción del sistema actual	2
1.3. Objetivos y alcance del proyecto	3
1.4. Estructura del Documento	3
2. Planificación	5
2.1. Metodología de desarrollo	5
2.1.1. Etapas	5
2.2. Planificación del proyecto	7
2.3. Costes	10
3. Estudio y desarrollo previo	11
3.1. Ingeniería Dirigida por Modelos (MDE)	11
3.2. Lenguaje de Dominio Específico (DSL)	11
3.3. OpenSim	12
3.4. OpenSim en la Universidad en Cádiz	13
3.4.1. Arquitectura	13
3.4.2. Juegos	15
4. Análisis del Sistema	19
4.1. Catálogo de actores	19
4.2. Requisitos funcionales	19
4.2.1. Caso de Uso: Consulta VWQL	20
4.2.2. Caso de uso: Evidencias generales	21
4.2.3. Caso de Uso: Evidencias Memory	23
4.2.4. Caso de Uso: Evidencias HiddenRoom	24
4.2.5. Caso de Uso: Evidencias Coffeeshop	24
4.3. Requisitos de información	25
4.3.1. Mundos Virtuales	25
4.3.2. Metamodelo	28
4.4. Requisitos no funcionales	30
4.5. Estudio de alternativas tecnológicas	31
5. Diseño del Sistema	33
5.1. Diseño de la arquitectura	33
5.2. Arquitectura física	34
5.3. Arquitectura lógica	34
5.3.1. Transformaciones Kettle	35

5.3.2.	Estructura fichero Kettle	42
5.3.3.	Ejemplo mapeo de VWQL a KTR de Kettle	43
5.4.	Arquitectura de diseño	46
5.5.	Diseño interfaz de usuario	47
6.	Implementación del Sistema	51
6.1.	Metamodelo	51
6.1.1.	Metamodelo etapa 2	53
6.1.2.	Metamodelo final	54
6.2.	Gramática	55
6.3.	Funcionalidad	58
6.4.	Control de versiones	59
6.5.	Herramientas usadas	60
6.5.1.	Eclipse Modeling Project (EMP)	60
6.5.2.	Eclipse Modeling Framework (EMF)	60
6.5.3.	Ecore Tools	60
6.5.4.	Xtext	61
6.5.5.	Xtend	62
6.5.6.	Kettle (Pentaho Data Integration)	62
6.5.7.	MySQL	66
6.5.8.	L ^A T _E X	66
6.5.9.	Draw.io	67
6.5.10.	GIMP	67
7.	Pruebas del Sistema	69
7.1.	Pruebas unitarias	69
7.1.1.	Metamodelo	69
7.1.2.	Editor	70
7.1.3.	Transformación	70
7.1.4.	Resultados	71
7.2.	Pruebas de integración	71
7.3.	Pruebas de sistema y aceptación	71
7.3.1.	Pruebas funcionales	71
8.	Conclusiones	75
8.1.	Objetivos	75
8.2.	Dificultades	76
8.2.1.	Desarrollo de aplicación	76
8.2.2.	Estudio e integración de Kettle	76
8.2.3.	Estudio e integración de diccionarios	77
8.2.4.	Facilidad de uso	77
8.3.	Experiencia personal	77
8.4.	Trabajo futuro	78
	Apéndices	79
A.	Manual de usuario	81
A.1.	Introducción	81
A.2.	Características	81
A.3.	Requisitos previos	82

A.4. Utilización	82
B. Manual de Instalación EvalSim	87
B.1. Requisitos previos	87
B.2. Procedimientos de instalación	87
C. Bases de Datos	89
C.1. Base de Datos: Avatar Tracking	89
C.1.1. Tabla <i>anke</i>	89
C.2. Base de Datos: UCA Cafeteria Version 2	90
C.2.1. Tabla <i>clientes</i>	90
C.2.2. Tabla <i>clientes_servidos</i>	90
C.2.3. Tabla <i>partida</i>	91
C.3. Base de Datos: UCA Memory Ordenar Habitacion	92
C.3.1. Tabla <i>sala_1</i>	92
C.3.2. Tabla <i>sala_2</i>	92
C.4. Base de Datos: UCA Coffeeshop	94
C.4.1. Tabla <i>cafeteria</i>	94
C.4.2. Tabla <i>casino</i>	95
C.4.3. Tabla <i>mango</i>	95
C.4.4. Tabla <i>quiz</i>	95
C.4.5. Tabla <i>saturn</i>	96
C.5. Base de Datos: UCA Memory Saturn	97
C.5.1. Tabla <i>sala_1</i>	97
C.6. Base de Datos: UCA Supermercado	98
C.6.1. Tabla <i>aktivitat2</i>	98
C.6.2. Tabla <i>aktivitat3</i>	99
C.6.3. Tabla <i>aktivitat4</i>	99
C.7. Base de Datos: ROBUST	100
C.7.1. Tabla <i>assets</i>	100
C.7.2. Tabla <i>auth</i>	101
C.7.3. Tabla <i>avatars</i>	102
C.7.4. Tabla <i>friends</i>	103
C.7.5. Tabla <i>griduser</i>	104
C.7.6. Tabla <i>inventoryfolder</i>	104
C.7.7. Tabla <i>inventoryitems</i>	105
C.7.8. Tabla <i>presence</i>	107
C.7.9. Tabla <i>regions</i>	107
C.7.10. Tabla <i>tokens</i>	108
C.7.11. Tabla <i>useraccounts</i>	108
C.8. Base de Datos: OPENSIM	109
C.8.1. Tabla <i>estateban</i>	109
C.8.2. Tabla <i>estate_groups</i>	109
C.8.3. Tabla <i>estate_managers</i>	109
C.8.4. Tabla <i>estate_map</i>	110
C.8.5. Tabla <i>estate_settings</i>	110
C.8.6. Tabla <i>estate_users</i>	110
C.8.7. Tabla <i>land</i>	110
C.8.8. Tabla <i>landaccesslist</i>	110
C.8.9. Tabla <i>migrations</i>	110

C.8.10. Tabla <i>primitems</i>	110
C.8.11. Tabla <i>primshapes</i>	110
C.8.12. Tabla <i>regionban</i>	110
C.8.13. Tabla <i>regionenvironment</i>	110
C.8.14. Tabla <i>regionextra</i>	111
C.8.15. Tabla <i>regionsettings</i>	111
C.8.16. Tabla <i>regionwindlight</i>	111
C.8.17. Tabla <i>spawn_point</i>	111
C.8.18. Tabla <i>terrain</i>	111
C.9. Base de Datos: OPENSIM MODIFICADO	112
C.9.1. Tabla <i>_actions</i>	112
C.9.2. Tabla <i>_avatar</i>	113
C.9.3. Tabla <i>_bournechannel</i>	113
C.9.4. Tabla <i>_bourneim</i>	114
C.9.5. Tabla <i>_events</i>	114
C.9.6. Tabla <i>_regions</i>	114
C.9.7. Tabla <i>_teleport</i>	115
C.9.8. Tabla <i>_trace</i>	115
D. Ejemplo de archivo KTR	117
E. Archivo metamodelo Ecore	165
F. Publicaciones	169
F.1. Artículo enviado a TEEM'15	170
F.2. Artículo enviado a JID2016	179
Bibliografía y referencias	183
GNU Free Documentation License	185
1. APPLICABILITY AND DEFINITIONS	185
2. VERBATIM COPYING	186
3. COPYING IN QUANTITY	186
4. MODIFICATIONS	187
5. COMBINING DOCUMENTS	188
6. COLLECTIONS OF DOCUMENTS	189
7. AGGREGATION WITH INDEPENDENT WORKS	189
8. TRANSLATION	189
9. TERMINATION	189
10. FUTURE REVISIONS OF THIS LICENSE	190
11. RELICENSING	190
ADDENDUM: How to use this License for your documents	190

Índice de figuras

2.1. Tabla con planificación del proyecto.	8
2.2. Diagrama de gantt, con la planificación del proyecto.	9
3.1. Sistema Antiguo.	14
3.2. Sistema Nuevo.	14
3.3. Captura de juego Memory	15
3.4. Captura juego HiddenRoom, vista jugador con mando.	16
3.5. Captura juego HiddenRoom, vista jugador con visor.	16
3.6. Captura juego Saturn, vista jugador con lista de la compra.	17
3.7. Captura juego Coffeeshop, jugador hablando con cliente-bot.	17
4.1. Diagrama de caso de uso del sistema.	19
4.2. Diagrama de super caso de uso del sistema.	21
4.3. Diagrama de caso de uso <i>Evidencias generales</i> del sistema.	21
4.4. Diagrama de caso de uso <i>Evidencias Memory</i> del sistema.	23
4.5. Diagrama de caso de uso <i>Evidencias HiddenRoom</i> del sistema.	24
4.6. Diagrama de caso de uso <i>Evidencias Coffeeshop</i> del sistema.	24
4.7. Diagrama de bases de datos de los mundos virtuales.	26
4.8. Metamodelo del EvalSim.	29
5.1. Diseño de la arquitectura lógica.	33
5.2. Diagrama de la arquitectura física actual de EvalSim.	34
5.3. Diagrama de la arquitectura lógica de EvalSim.	35
5.4. Imagen 1 de transformación de ejemplo con EvalSim.	44
5.5. Imagen 2 de transformación de ejemplo con EvalSim.	45
5.6. Diagrama de arquitectura de diseño.	47
5.7. Captura del wizard de selección del proyecto.	48
5.8. Captura de la estructura inicial del proyecto.	48
5.9. Captura de Evalsim.	49
5.10. Captura de la estructura del proyecto después de ejecutar una consulta.	49
6.1. Paso 1: creación proyecto EMP.	52
6.2. Paso 2: dar nombre y ubicación al proyecto.	52
6.3. Paso 3: estructura del proyecto y editor visual de Sirius.	53
6.4. Diagrama del metamodelo en la etapa 2.	53
6.5. Diagrama de la versión definitiva del metamodelo.	54
6.6. Contenido del fichero Ecore.	55
6.7. Estructura final del proyecto.	59
6.8. Jerarquía de componentes Ecore.	61
7.1. Prueba unitaria del metamodelo.	70

A.1. Selección proyecto EvalSim o VWQL.	83
A.2. Jerarquía de componentes Ecore.	83
A.3. Jerarquía de componentes Ecore.	84
A.4. Jerarquía de componentes Ecore.	85
A.5. Jerarquía de componentes Ecore.	85
A.6. Jerarquía de componentes Ecore.	86
A.7. Jerarquía de componentes Ecore.	86
B.1. Carpeta plugins dentro de carpeta Eclipse.	88
C.1. Diagrama ER de la base de datos <i>avatar-tracking</i>	89
C.2. Diagrama ER de la base de datos <i>uca_cafeteriaversion2</i>	90
C.3. Diagrama ER de la base de datos <i>uca_memory_ordenar_habitacion</i>	92
C.4. Diagrama ER de la base de datos <i>uca_coffeeshop</i>	94
C.5. Diagrama ER de la base de datos <i>uca_memory_saturn</i>	97
C.6. Diagrama ER de la base de datos <i>uca_supermercado</i>	98
C.7. Diagrama ER de la base de datos <i>robust</i>	100
C.8. Diagrama ER de la base de datos <i>os_</i>	109
C.9. Diagrama ER de la base de datos <i>os_modificada</i>	112

Índice de tablas

2.1. Cómputo coste estimado del proyecto.	10
7.1. Pruebas unitarias del editor.	70
7.2. Pruebas unitarias de transformación.	71
7.3. Pruebas de integración de los módulos.	71
7.4. Pruebas de funcionales, resultado de consulta <i>prepositions</i>	73
7.5. Pruebas de funcionales, resultado de consulta <i>checks</i>	74
C.1. Tabla <i>anke</i> de la BD <i>avatar-tracking</i>	89
C.2. Tabla <i>clientes</i> de la BD <i>uca_cafeteriaversion2</i>	90
C.3. Tabla <i>clientes_servidos</i> de la BD <i>uca_cafeteriaversion2</i>	91
C.4. Tabla <i>partida</i> de la BD <i>uca_cafeteriaversion2</i>	91
C.5. Tabla <i>sala_1</i> de la BD <i>uca_memory_ordenar_habitacion</i>	92
C.6. Tabla <i>sala_2</i> de la BD <i>uca_memory_ordenar_habitacion</i>	93
C.7. Tabla <i>cafeteria</i> de la BD <i>uca_coffeeshop</i>	95
C.8. Tabla <i>casino</i> de la BD <i>uca_coffeeshop</i>	95
C.9. Tabla <i>mango</i> de la BD <i>uca_coffeeshop</i>	95
C.10. Tabla <i>quiz</i> de la BD <i>uca_coffeeshop</i>	96
C.11. Tabla <i>saturn</i> de la BD <i>uca_coffeeshop</i>	96
C.12. Tabla <i>sala_1</i> de la BD <i>uca_memory_saturn</i>	97
C.13. Tabla <i>aktivitat2</i> de la BD <i>uca_supermercado</i>	99
C.14. Tabla <i>aktivitat3</i> de la BD <i>uca_supermercado</i>	99
C.15. Tabla <i>aktivitat4</i> de la BD <i>uca_supermercado</i>	99
C.16. Tabla <i>assets</i> de la BD <i>robust</i>	101
C.17. Tabla <i>auth</i> de la BD <i>robust</i>	102
C.18. Tabla <i>avatars</i> de la BD <i>robust</i>	103
C.19. Tabla <i>friends</i> de la BD <i>robust</i>	104
C.20. Tabla <i>griduser</i> de la BD <i>robust</i>	104
C.21. Tabla <i>inventoryfolder</i> de la BD <i>robust</i>	105
C.22. Tabla <i>inventoryitems</i> de la BD <i>robust</i>	106
C.23. Tabla <i>presence</i> de la BD <i>robust</i>	107
C.24. Tabla <i>regions</i> de la BD <i>robust</i>	108
C.25. Tabla <i>tokens</i> de la BD <i>robust</i>	108
C.26. Tabla <i>useraccounts</i> de la BD <i>robust</i>	108
C.27. Tabla <i>_actions</i> de la BD <i>opensim</i>	113
C.28. Tabla <i>_avatar</i> de la BD <i>opensim</i>	113
C.29. Tabla <i>_bournechannel</i> de la BD <i>opensim</i>	113
C.30. Tabla <i>_bourneim</i> de la BD <i>opensim</i>	114
C.31. Tabla <i>_events</i> de la BD <i>opensim</i>	114
C.32. Tabla <i>_regions</i> de la BD <i>opensim</i>	114

C.33. Tabla <i>_teleport</i> de la BD <i>opensim</i>	115
C.34. Tabla <i>_trace</i> de la BD <i>opensim</i>	115

Índice de código

5.1. Estructura de archivo XML con extensión <i>.ktr</i> de transformación Kettle.	42
5.2. Ejemplo de consulta de EvalSim en lenguaje VWQL.	43
5.3. Ejemplo de resultado de consulta en EvalSim.	46
6.1. Lenguaje VWQL completo	56
6.2. Gramática del lenguaje VWQL.	56
7.1. Consulta VWQL.	72
7.2. Consulta VWQL.	73
A.1. Lenguaje VWQL completo	84
archivos/ejemplo.ktr	117
archivos/modelo.ecore	165

Capítulo 1

Introducción

En la introducción pasaremos a describir la motivación del presente proyecto, la descripción del sistema actual, los objetivos y la estructura del documento del presente documento.

1.1. Motivación

En los últimos años muchas instituciones educativas han basado su docencia en procesos basados en *Blended Learning*, que combina la docencia presencial con un alto porcentaje de aprendizaje autónomo y tutorización on-line. De ahí la necesidad por parte de los profesores de desarrollar nuevas herramientas de aprendizaje a fin de facilitarle a los alumnos el proceso de aprendizaje autónomo y de garantizar que adquieran las competencias esperadas.

Una de las herramientas de aprendizaje autónomo on-line usadas, desde hace años en *Blended Learning* enfocada a la enseñanza de idiomas, son los *mundos virtuales* [Palomo-Duarte, M. et al., 2015a], [Molka-Danielsen, J. and Deutschmann, M., 2009], [Svensson, 2003]. Se entiende por mundo virtual un sistema de computación usado para crear un mundo artificial donde el usuario, a través de una representación gráfica llamada avatar, tiene la posibilidad de interactuar y manipular objetos dentro del entorno. Lo que hace particularmente atractivos a los mundos virtuales es la similitud con la apariencia del mundo real, junto con las diversas formas en la que se puede interactuar con los objetos y con los avatares. La interacción entre avatares (entre estudiantes, o entre estudiantes y profesores) se realiza en tiempo real y permite que se hagan actividades de forma cooperativa. En los mundos virtuales se pueden reproducir, simular o recrear situaciones del mundo real que introducen al usuario dentro de este mundo. [Berns, A. et al., 2013a]

Aunque los mundos virtuales presentan muchas ventajas, su uso no se ha extendido demasiado debido a diversas causas como su esfuerzo de desarrollo, las complicaciones de su administración y gestión, los requisitos tecnológicos necesarios para su ejecución, el escaso valor que tienen determinadas experiencias al repetirse y la problemática de su evaluación. [Palomo-Duarte, M. et al., 2015a]

La investigadora Anke Berns, junto a otros investigadores de la Universidad Autónoma de Madrid (UAM), comenzaron a explorar las posibilidades de fusionar mundos virtuales y videojuegos como herramienta para el aprendizaje autónomo on-line de los alumnos [Berns, A. et al., 2013a], [Berns, A. et al., 2011a], [Berns, A. et al., 2011b]. La razón es que los mundos virtuales, junto con los videojuegos, representan un estimulante entorno de aprendizaje, que incrementa significativamente la motivación de los estudiantes así como su formación.

Continuando con los estudios en el ámbito de los mundos virtuales, la investigadora Anke Berns en

conjunto con los investigadores Manuel Palomo y Juan M. Dodero del Departamento de Ingeniería Informática de la Universidad de Cádiz (UCA) entre otros, implantaron una plataforma de Mundos Virtuales con *OpenSim*. La plataforma estaba dirigida a alumnos de Alemán nivel A1 de la Facultad de Filosofía y Letras de la Universidad de Cádiz y en ella se desarrollaron e implementaron juegos de rol cooperativos y competitivos con el fin de reforzar las competencias comunicativas de los alumnos. [Palomo-Duarte, M. et al., 2015b], [Palomo-Duarte, M. et al., 2015a], [Berns, A. et al., 2013b], [Berns, A. et al., 2013c], [Berns, A. and Palomo-Duarte, M., 2012]

Mientras los alumnos jugaban dentro de los mundos virtuales, se iban almacenando datos de interacción de sus avatares con el mundo virtual y con otros avatares, tales como chat, posicionamiento dentro de las regiones virtuales, puntuación, tiempo de juego, etc. Estos datos posteriormente son analizados por los investigadores para realizar publicaciones con los resultados obtenidos. [Palomo-Duarte, M. et al., 2015b], [Palomo-Duarte, M. et al., 2015a], [Berns, A. et al., 2013b], [Berns, A. et al., 2013c], [Berns, A. and Palomo-Duarte, M., 2012]

La labor de extracción y procesamiento de datos es un trabajo largo y tedioso, que en algunas ocasiones, necesitan personas ajenas al mundo técnico informático. Es ahí cuando surge la idea de este proyecto, facilitar a los investigadores la labor de extracción de datos y que sea tan sencillo que lo pueda usar personal no cualificado técnicamente en extracción de datos de bases de datos (BBDD), ni en su posterior procesamiento.

El proyecto que se ha desarrollado va dirigido a facilitar las labores de investigación del personal docente de la UCA. En concreto, la extracción de indicadores en las BBDD de la plataforma de mundos virtuales desplegada en la Universidad, a través de un lenguaje de consulta muy sencillo.

La herramienta que se ha desarrollado es EvalSim, y el lenguaje de consulta es Virtual Worlds Query Language (VWQL). A través de la herramienta se pueden obtener **indicadores** de, entre otras cosas, la interacción que han tenido los alumnos a través del chat. Para ello se ha estado trabajando estrechamente con los investigadores a los que se dirige la herramienta. Esto nos ha permitido conocer que **indicadores** son, estimar cómo se pueden medir, que es importante valorar de lo que han hecho los estudiantes, y ver cómo puede estar plasmado todo eso en la información contenida en la base de datos para aprovecharlo.

Los datos que se puedan tomar o la adaptación de estas herramientas para ser utilizada con otros mundos virtuales, son campos que quedan abierto para futuros trabajos, y que serán explicados en mayor detalle en la sección 8.4 del capítulo Conclusiones.

1.2. Descripción del sistema actual

En la Univesidad de Cádiz ya se han desarrollado lenguajes y herramientas similares a la que se propone en este proyecto, como *EvalCourse*, aunque nunca enfocado a la obtención de indicadores en *mundos virtuales*.

EvalCourse es un sistema software que apoya procedimientos de evaluación de procesos de aprendizaje desarrollados en plataformas *LMS* (*Learning Management System*) [Balderas, A. et al., 2015], [Balderas, A. et al., 2014]. Ejecuta consultas escritas en un lenguaje de consultas específico de evaluación (*SASQL* - *Assesment-Specific Query Language*). *SASQL* es un *DSL* para obtener indicadores objetivos orientados

a la evaluación. El sistema esta principalmente pensado para *Moodle* por ser la plataforma que se usa en la Universidad de Cádiz, y , en comparación con otros sistemas y herramientas similares, extrae indicadores relevantes para más competencias y para más formas de trabajo de los alumnos desde wikis, tareas, foros, talleres...

Esta herramienta sirvió como punto de partida para desarrollar el sistema de evaluación de *mundos virtuales EvalSim*. *EvalSim* presenta una sintáxis parecida y el funcionamiento interno es bastante similar, aunque presenta bastantes diferencias debido a que el campo al que van dirigidos cada uno de los sistemas es totalmente diferente. Mientras que *EvalCourse* va dirigido a la obtención de indicadores de competencias y formas de trabajo de los alumnos a través del LMS *Moodle*, *EvalSim* va enfocado a la obtención de indicadores en el aprendizaje autónomo de idiomas de los alumnos a través de actividades en mundos virtuales. Otra diferencia que presenta es que las herramientas usadas para la generación de *EvalSim* han sido actualizadas y en algunas casos, hasta han cambiado completamente la forma en que se hacia.

1.3. Objetivos y alcance del proyecto

Objetivo principal

- **Creación de DSL.** Crear un *DSL* que extraiga información útil e importante para el investigador sobre la interacción y comportamiento de los alumnos en los mundos virtuales.
- **Facilidad de uso.** Hacer que el software sea muy fácil de usar, de forma que cualquier usuario pueda utilizarlo con sencillos pasos.

Objetivos secundarios

- **Sintáxis sencilla.** El *DSL* debe tener una sintáxis tan sencilla e intuitiva que con solo una pequeña guía, cualquier investigador pueda usarla.
- **Metamodelo genérico.** Desarrollar un metamodelo lo más genérico posible para el *DSL*, de forma que sea válido para posibles expansiones del mundo virtual y admita la integración de nuevos indicadores.
- **Investigar indicadores.** Estudiar y hablar con profesores del idioma en cuestión (alemán) para ver que información puede ser más importante analizar y generar consultas que extraigan y procesen los datos para que el sistema nos devuelva dicha información
- **Investigar *Kettle* de Pentaho.** Investigar el funcionamiento de *Kettle* para generar las consultas de los indicadores y estudiar la estructura del fichero, para que el *DSL* pueda generar un fichero válido para la ejecución en *Kettle*.
- **Integración del Software.** Integrar el software como un todo: Eclipse, EvalSim y Kettle para facilitar al usuario la implantación del software.

1.4. Estructura del Documento

Este documento se divide en nueve capítulos en total y algunos apéndices:

- En el capítulo 1, se realiza una pequeña introducción al proyecto, mostrando de donde ha surgido la idea de EvalSim.
- En el capítulo 2, se muestra la planificación que se ha seguido en el desarrollo del proyecto, junto con la metodología utilizada y un cálculo estimado del coste del proyecto.
- En el capítulo 3, se ha hecho un pequeño resumen previo a la explicación del proyecto, que hará que se entienda mejor lo que se describe en esta memoria.
- En el capítulo 4, se explica la fase de análisis que se ha realizado para el desarrollo de EvalSim.
- En el capítulo 5, se expone la fase de diseño del sistema con detalle.
- En el capítulo 6, se explica la implementación del DSL y de la infraestructura desarrollada en la que se va a ejecutar.
- En el capítulo 7, se detallan las pruebas realizadas para comprobar el correcto funcionamiento del sistema.
- En el capítulo 8, se exponen las conclusiones que se han sacado del desarrollo del proyecto, una vez finalizado.
- En los apéndices, se anexan el manual de usuario, el manual de instalación, los artículos enviados a congresos y diferentes documentos que aportan una visión más general del proyecto.

Capítulo 2

Planificación

En esta sección se describen todos los aspectos relativos a la planificación del proyecto: metodología, organización, costes y planificación.

2.1. Metodología de desarrollo

En el desarrollo de la herramienta EvalSim se ha seguido una metodología iterativa incremental. Esta metodología de desarrollo nos permite crear versiones del producto poco a poco.

Como principales ventajas podemos destacar que no es necesario esperar hasta el final del proceso para obtener una versión del producto, ni tampoco todos los requisitos deben estar definidos al comienzo, tras cada iteración pueden ser redefinidos.

Esta metodología es ideal para proyectos en los que no se sabe desde un primer momento hasta qué punto y cómo puede realizarse el diseño del problema. De esta forma, se pueden ir cumpliendo pequeños hitos que nos irán marcando el camino a seguir.

2.1.1. Etapas

En este apartado se explicará en que etapas se ha dividido el proyecto y que se ha hecho en cada etapa.

Etapas 1: Tecnologías

Como el sistema y las tecnologías son complejos, durante la primera etapa se realizó un primer acercamiento a las tecnologías que se van a utilizar para el desarrollo del proyecto. Para ello se generó un primer prototipo con un metamodelo y una gramática sumamente sencillos, que generaban un fichero vacío con extensión **.ktr** de Kettle. También en este ciclo se empiezan a estudiar las herramientas que van a ser usadas Eclipse Modeling Framework (EMF), Ecore, Sirius, Xtext.

- Estudio tecnologías
- Creación de metamodelo Ecore, generado con Sirius
- Creación de gramática
- Creación del proyecto Xtext

Etapla 2: Estudio de requisitos y primera versión

En la segunda etapa se comienza a estudiar las herramientas Xtend y Kettle. Se plantean los primeros requisitos del sistema y se genera una primera versión del metamodelo con su correspondiente gramática.

1. Estudio de las BBDD de OpenSim y primeros requisitos
2. Estudio tecnologías
3. Creación de consulta estática con Kettle a las BBDD
4. Actualización del metamodelo Ecore (se considera la primera versión)
5. Actualización de la gramática (se considera la primera versión)
6. Actualización del proyecto Xtext
7. Programación del DSL en Xtend (primera consulta de forma estática)

Etapla 3: Mejora de metamodelo y consulta dinámica

En esta etapa se cambia el metamodelo para que admita la posibilidad de añadir juegos futuros sin que los cambios sean demasiado costosos y se realizan cambios y mejoras en el sistema. Se genera la primera consulta de forma dinámica.

- Estudio de nuevos requisitos
- Actualización del metamodelo Ecore
- Actualización de la gramática
- Actualización del proyecto Xtext
- Actualización del DSL con Xtend (generación de consulta dinámica con conexión a BBDD)
- Pruebas

Etapla 4: Consultas dinámicas

En la etapa 4 se concentra toda la parte de generación de consultas de forma dinámica. Para algunas consultas son necesarios diccionarios, que son generados en esta etapa. También se realizan cambios en el metamodelo y gramática.

- Estudio de nuevos requisitos
- Actualización del metamodelo Ecore
- Actualización de la gramática
- Actualización del proyecto Xtext
- Actualización del DSL con Xtend (generación de consultas con diccionario)
- Pruebas

Etapas 5: Mejoras y personalización

En esta etapa se realizan algunas mejoras en el DSL y se integran las bibliotecas de Pentaho. También se añade una opción en el menú del editor que genera y ejecuta el archivo Kettle con la consulta

- Actualización del proyecto Xtext
- Actualización del DSL con Xtend (mejora de las consultas con diccionario)
- Pruebas
- Creación de menú para generar y ejecutar

Etapas 6: Finalización y exportación

En la última etapa se refactoriza el código y se exporta el DSL como plugin.

- Refactorización del código
- Exportación del plugin
- Pruebas

2.2. Planificación del proyecto

A continuación se procede a exponer la organización temporal que se ha seguido para el desarrollo del proyecto.

Dentro de la tabla de la planificación (Figura 2.1) aparecen las etapas descritas en la sección anterior (Subsección 2.1.1). En esta tabla se detallan la etapa con cada una de sus tareas, la fecha de inicio, la fecha de finalización y el número de días que han transcurrido entre el inicio y el fin de la tarea.

En el diagrama de Gantt (Figura 2.2) se detallan, al igual que en la tabla de planificación (Figura 2.1), las etapas y tareas en las que se ha dividido el proyecto. También se detalla el transcurso de tiempo desde el inicio del proyecto, aunque de una forma más visual y fácil de entender. Por ejemplo, hay tareas que se han ido haciendo en paralelo tal como se puede apreciar en este diagrama, y que en la tabla de planificación no se aprecia de forma tan clara.

WBS	Nombre	Inicio	Fin	Trabajo	Duración
1	Etapa 1	dic 1	ene 10	48d	29d
1.1	Estudio tecnologías E1	dic 1	dic 28	20d	20d
1.2	Creación metamodelo	dic 16	dic 27	8d	8d
1.3	Creación gramática	dic 28	ene 10	10d	10d
1.4	Creación proyecto Xtext	dic 28	ene 10	10d	10d
2	Etapa 2	ene 11	mar 1	45d	36d
2.1	Estudio tecnologías E2	ene 11	ene 31	15d	15d
2.2	Creación consulta Kettle	ene 23	ene 27	5d	5d
2.3	Estudio BBDD y requisitos	feb 1	feb 9	7d	7d
2.4	Actualización metamodelo E2	feb 10	feb 16	5d	5d
2.5	Actualización gramática E2	feb 17	feb 23	5d	5d
2.6	Actualización proyecto Xtext	feb 24	mar 1	4d	4d
2.7	Programación DSL con Xtend	feb 24	mar 1	4d	4d
3	Etapa 3	mar 2	mar 27	32d	18d
3.1	Estudio requisitos E3	mar 2	mar 3	2d	2d
3.2	Actualización metamodelo E3	mar 6	mar 9	4d	4d
3.3	Actualización gramática E3	mar 6	mar 9	4d	4d
3.4	Actualización proyecto Xtext E3	mar 10	mar 23	10d	10d
3.5	Actualización DSL con Xtend E3	mar 10	mar 23	10d	10d
3.6	Pruebas E3	mar 24	mar 27	2d	2d
4	Etapa 4	mar 28	jun 7	92d	52d
4.1	Estudio de requisitos E4	mar 28	abr 3	5d	5d
4.2	Actualización metamodelo E4	abr 4	abr 4	1d	1d
4.3	Actualización gramática E4	abr 5	abr 5	1d	1d
4.4	Actualización proyecto Xtext E4	abr 6	may 31	40d	40d
4.5	Actualización DSL con Xtend E4	abr 6	may 31	40d	40d
4.6	Pruebas E4	jun 1	jun 7	5d	5d
5	Etapa 5	jun 8	jul 21	52d	32d
5.1	Actualización proyecto Xtext E5	jun 8	jul 5	20d	20d
5.2	Actualización DSL con Xtend E5	jun 8	jul 5	20d	20d
5.3	Pruebas E5	jul 6	jul 7	2d	2d
5.4	Creación menú para ejecutar	jul 10	jul 21	10d	10d
6	Etapa 6	jul 24	ago 2	8d	8d
6.1	Refactorización del código	jul 24	jul 26	3d	3d
6.2	Exportación plugin	jul 27	jul 31	3d	3d
6.3	Pruebas E6	ago 1	ago 2	2d	2d

Figura 2.1: Tabla con planificación del proyecto.

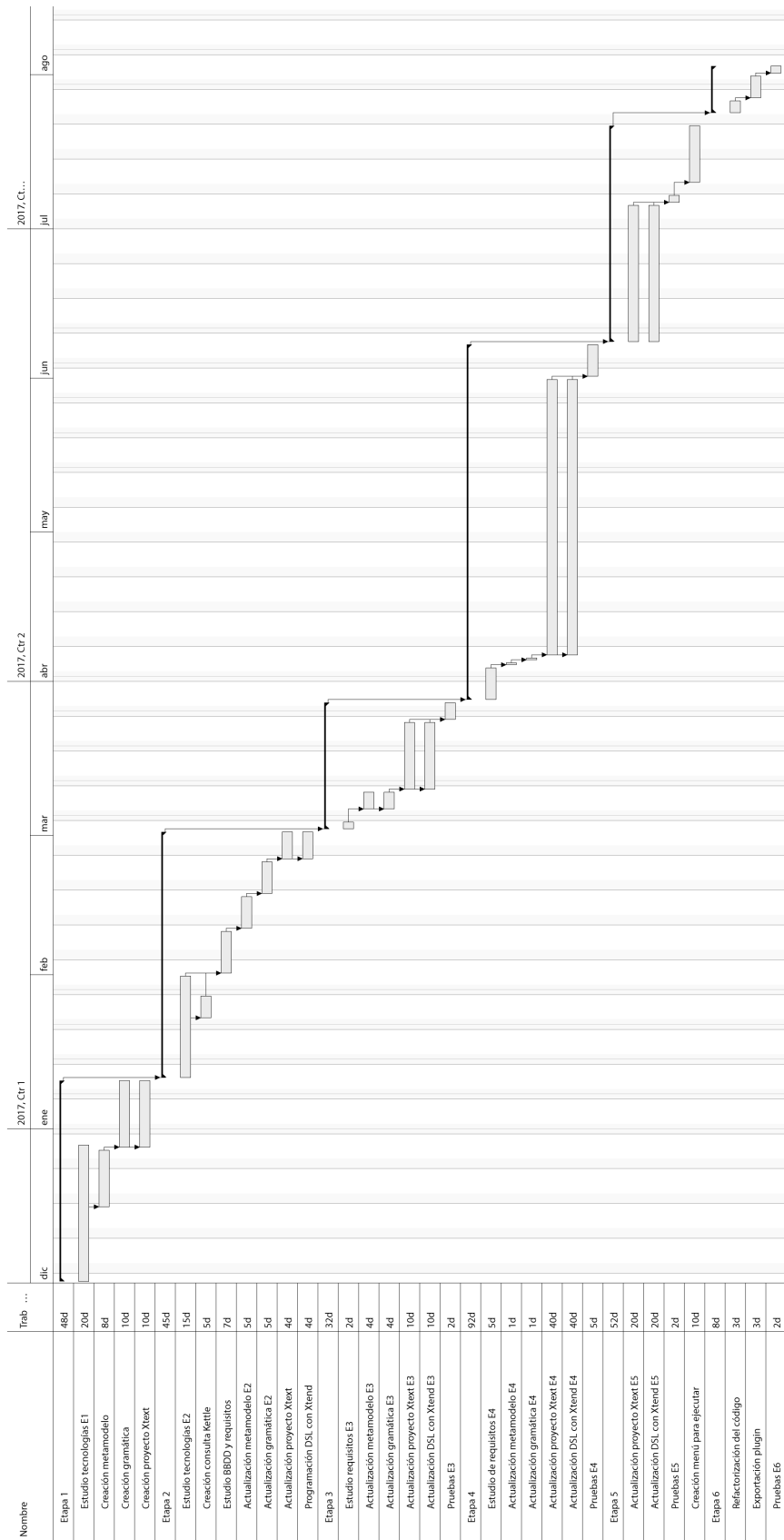


Figura 2.2: Diagrama de gantt, con la planificación del proyecto.

2.3. Costes

A continuación se expondrá el presupuesto necesario para llevar a cabo el presente proyecto (Tabla 2.1). Para realizar el mismo, nos basaremos en los salarios base que la UCA aplica al personal técnico de apoyo laboral contratado (CCOO [2010]).

Para el cómputo del coste total, se tendrá en cuenta que el proyecto ha sido realizado por una sola persona.

Categoría	T.S. Apoyo a la Docencia e Investigación
Salario base mensual	1606,15 €
Complemento productividad	834,82 €
Tiempo estimado	8 meses
Total estimado	$8 \times (1606,15 + 834,82) = 19527,76\text{€}$

Tabla 2.1: Cómputo coste estimado del proyecto.

Capítulo 3

Estudio y desarrollo previo

En este capítulo daremos un acercamiento teórico a la tecnología que existe actualmente y que se va a usar y con la que se va a desarrollar el proyecto de EvalSim.

3.1. Ingeniería Dirigida por Modelos (MDE)

MDE es un nuevo enfoque en Ingeniería del Software. En él se utilizan modelos como artefactos software. Es una metodología de desarrollo de software que se centra en la creación y explotación de modelos de dominio (es decir, representaciones abstractas de los conocimientos y las actividades que rigen en un dominio de aplicación particular), y no en la conceptos o algoritmos informáticos.

El enfoque que aporta MDE tiene por objeto aumentar la productividad mediante la maximización de la compatibilidad entre los sistemas (a través de la reutilización de modelos estandarizados), simplificando el proceso de diseño (a través de modelos de patrones de diseño que se repiten en el dominio de aplicación), y promoviendo la comunicación entre los individuos y equipos que trabajan en el sistema (a través de una estandarización de la terminología y las mejores prácticas utilizadas en el dominio de aplicación).

Un paradigma de modelado para el MDE se considera eficaz si los modelos tienen sentido desde el punto de vista de un usuario que está familiarizado con el dominio, y si ellos pueden servir como base para la implementación de sistemas. El modelo se debe desarrollar a través de una amplia comunicación entre los jefes de producto, diseñadores, desarrolladores y usuarios del dominio de aplicación. A medida que los modelos se acercan a la terminación, que permiten el desarrollo de software y sistemas.

3.2. Lenguaje de Dominio Específico (DSL)

Un DSL es un lenguaje de programación dedicado a un problema de dominio en particular. No tienen la intención de proporcionar características para resolver todo tipo de problemas.

Crear un DSL, junto con el software que lo soporte, puede valer la pena si el lenguaje permite expresar tipos de problemas y soluciones particulares que los lenguajes existentes no pueden modelar tan fácilmente.

Es posible que con un DSL no podamos implementar todos los programas que se pueden implementar con, por ejemplo, Java o C, que son conocidos como lenguajes de propósito general. Por otro lado, si el dominio del problema está cubierto por un DSL, se podrá solucionar ese problema de forma más rápida

y sencilla que con un lenguaje de propósito general. [Bettini, 2016]

Los DSL presentan una serie de ventajas:

- Los DSL permiten expresar soluciones usando los términos y el nivel de abstracción apropiado para el dominio del problema. En consecuencia, los mismos expertos de dominio pueden comprender, validar, modificar y a menudo desarrollar programas en DSL.
- Es código auto-documentado.
- Los DSL mejoran la calidad, productividad, confianza, mantenibilidad, portabilidad y reusabilidad de las aplicaciones.
- Los DSL permiten validaciones a nivel del dominio. Mientras las construcciones del lenguaje estén correctas, cualquier sentencia escrita puede considerarse correcta.

En contrapartida, los DSL presentan las siguientes desventajas:

- El coste de aprender un nuevo lenguaje vs. su aplicación limitada.
- El coste de diseñar, implementar y mantener un DSL y las herramientas para trabajar con él.
- Encontrar, establecer y mantener el alcance adecuado.
- Potencial pérdida de eficiencia y rendimiento en comparación con el software desarrollado de forma no automatizada.

3.3. OpenSim



El proyecto *OpenSim* (*Open Simulator*) fue fundado en enero de 2007 por Darren Guard (también conocido como MW), que, como tantas otras personas, vió el potencial de un servidor de entornos virtuales 3D de código abierto que podría ser utilizado para muchas aplicaciones diferentes. Al igual que muchos otros, Darren había visto otros intentos fallidos de servidores de mundos virtuales de código abierto, a menudo debido a la enorme tarea de escribir tanto un servidor como un cliente al mismo tiempo.

Luego, en enero de 2007, el cliente de *Second Life(TM)* fue liberado como código abierto, y *libsl* (una biblioteca BSD de código abierto para crear clientes personalizados que podrían conectarse a *Second Life(TM)*), estaba llegando al punto de ser estable. Así nació la idea de *OpenSimulator*, con el objetivo inicial de probar que el concepto del servidor cliente *SL* podría conectarse y permitir algunas funciones básicas. Lo que se espera es que con el tiempo el alcance del proyecto pueda llegar a ser mucho mayor que en sus humildes comienzos. Esto ha ocurrido, con el objetivo actual de elaborar una plataforma

estándar de entorno virtual que cualquier aplicación puede utilizar como marco. A pesar de que todavía mantienen la compatibilidad con el cliente de *Second Life*, han estado trabajando para apoyar a otros clientes. En el futuro se espera también soportar los protocolos y entornos que son completamente independientes de los de *Second Life*.

OpenSim es un desarrollo conjunto de código libre que tiene por objeto generar servidores de metaversos como *Second Life*. Mientras que *Second Life* es privado, tiene una economía real y su framework es cerrado, *OpenSim* es todo lo contrario, está implementado en C# y puede ser ejecutado tanto en Windows sobre .NET Framework como en Linux sobre Mono Framework. Se puede conseguir el código fuente que está amparado por una licencia tipo BSD y se puede usar en productos comerciales.

OpenSim tiene como finalidad crear entornos virtuales 3D con diferentes propósitos como educación, marketing, simulación, entretenimiento, etc. y todo en tiempo real.

La razón por la que se eligió *OpenSim* es que es un software de código abierto que se puede utilizar, con los módulos de software apropiados, como una plataforma de e-Learning o LMS. Para utilizar *OpenSim* como una plataforma de e-Learning se diseñó VirtUAM. A continuación se muestran las características que posee *OpenSim* frente a otras plataformas similares:

- El espacio virtual, que puede ser construido por sus usuarios y el número de prims (formas primitivas) creados en esta plataforma son ilimitados. Sin embargo, un gran número de prims, scripts o sims tiene un efecto negativo en el rendimiento del sistema.
- Por lo general, las plataformas de mundos virtuales son lugares públicos virtuales que no pueden ser completamente configurados para proporcionar un ambiente educativo exclusivamente controlada por sus usuarios, es decir, que cualquier persona podría acceder e interferir en el desarrollo de las actividades.
- La información relacionada con la conducta de los estudiantes, como estudiante-profesor y la interacción alumno-alumno o los registros de chat no pueden ser fácilmente recuperados desde plataformas externas a los mundos virtuales.
- *OpenSim* es un software de código abierto, que permite a los administradores y profesores modificar el programa cuando lo deseen. Tales modificaciones podrían tener como objetivo el almacenamiento de la conducta del jugador dentro de un sistema de base de datos o, incluso la inclusión de un mecanismo, que puede detectar problemas individuales de alumnos, a fin de centrarse específicamente en estos.

3.4. OpenSim en la Universidad en Cádiz

En esta sección veremos la plataforma de mundos virtuales que existe montada en la Universidad de Cádiz. Presentaremos la arquitectura física que existía junto con la que queremos implantar, y veremos como son y en que consisten los juegos desarrollados en los mundos virtuales.

3.4.1. Arquitectura

Ya hemos mencionado que para el proceso de investigación docente, era necesario disponer de un técnico o administrador que extrajese los datos de la base de datos y los procesase.

En la Figura 3.1 podemos observar como estaba dispuesta la arquitectura. En ella se puede ver que, por un lado, existe un servidor que permite que los alumnos se conecten a los mundos virtuales, e interactúen

dentro de ellos generando datos. Los datos generados se van almacenando a su vez en las bases de datos a través del servidor.

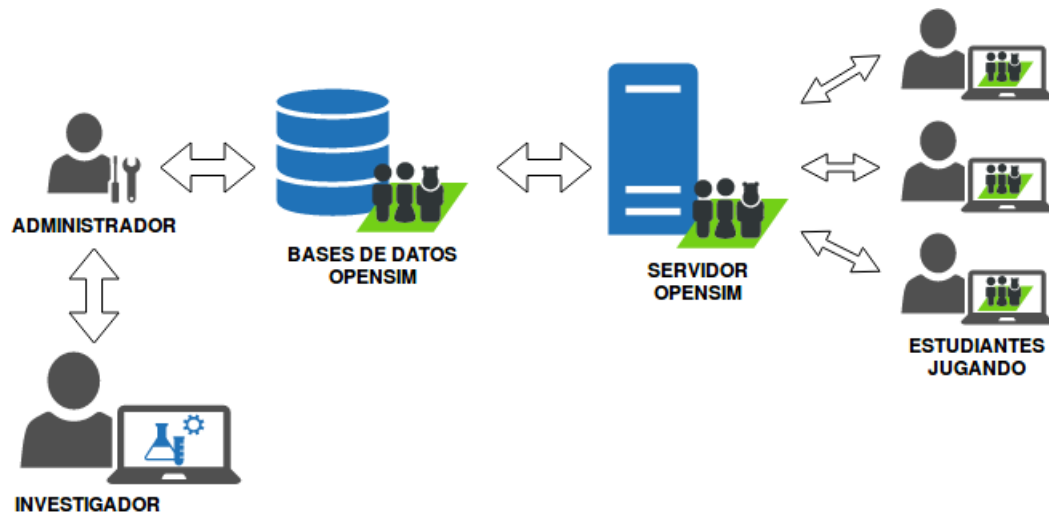


Figura 3.1: Sistema Antiguo.

Por otro lado podemos ver la parte de análisis e investigación. Tenemos un administrador o técnico informático que se encarga de comunicarse con las bases de datos, realizando consultas. Posteriormente los datos extraídos son procesados por el administrador/técnico. Estos datos se envían al docente o investigador, que se encargará de analizar y estudiar los datos para extraer la información útil.

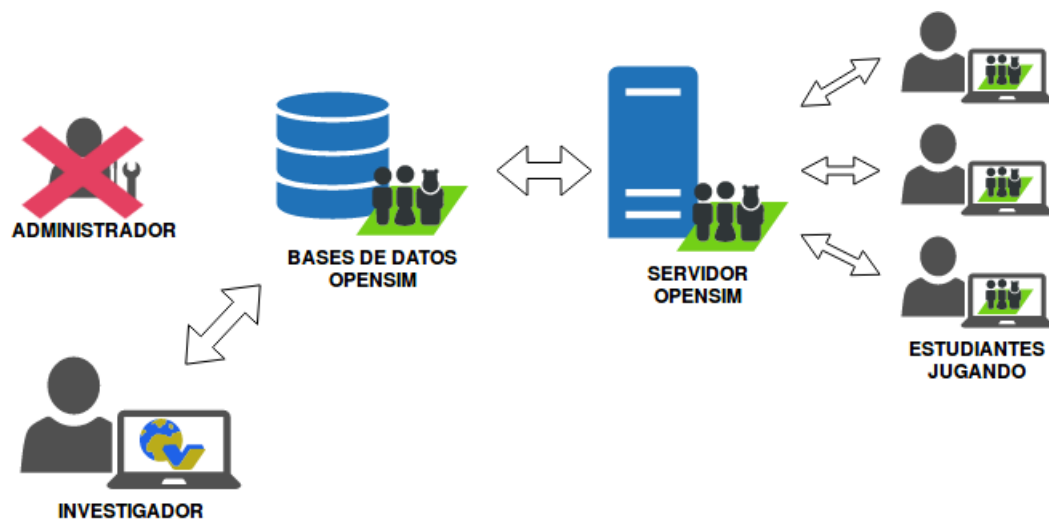


Figura 3.2: Sistema Nuevo.

La arquitectura (Figura 3.2) que se va a construir con EvalSim, pretende eliminar al administrador/técnico. De esta forma ahorramos tiempo, y automatizamos el proceso de extracción de indicadores útiles para investigación.

3.4.2. Juegos

En la Universidad de Cádiz, con algunos aportes de la Universidad Autónoma de Madrid, se han desarrollado varias actividades/juegos dentro de los mundos virtuales OpenSim. En esta subsección pasaremos a describir los juegos y en que consisten.

Memory

Los juegos Memory (Figura 3.3) se desarrollan en una sala de una región virtual, que tiene varios paneles en los que sentarse y jugar. Este juego es introductorio para otros juegos, ya que sirve para refrescar el vocabulario al jugador. Existen dos juegos Memory, uno aplicado a vocabulario del juego HiddenRoom y otro al vocabulario del juego Saturn.

El juego básicamente consiste en emparejar una imagen del lado izquierdo, que contiene un objeto o situación entre objetos; con la imagen correspondiente del lado derecho, que contiene un texto escrito en alemán con la descripción de un objeto o de una situación entre objetos. Al seleccionar una imagen del lado derecho se reproduce un audio con la pronunciación en alemán del texto de la imagen.



Figura 3.3: Captura de juego Memory

HiddenRoom

El juego HiddenRoom (Figuras 3.4 y (Figura 3.5)) se desarrolla en una casa situada en una región del mundo virtual. Es un juego por parejas y colaborativo, en el que se usa el chat para comunicarse con el compañero.

Es un juego de rol en el cual ambos jugadores tienen que ordenar los objetos de una habitación, previamente desordenada por el programa. Para ello un jugador A se le asigna un mando virtual (Figura 3.4), que le permite mover los objetos de la habitación y colocarlos en el sitio que le indica otro jugador B. Al jugador B, a su vez, se le asigna un visor con la posición correcta de los objetos (Figura 3.5).

Durante toda la actividad los jugadores tienen que colaborar y comunicarse vía chat, asegurándose de que los objetos están en su posición correcta. Una vez terminada la actividad el programa indica a los

jugadores los objetos acertados, o bien los no acertados, y la puntuación conseguida, a lo largo de la partida.



Figura 3.4: Captura juego HiddenRoom, vista jugador con mando.



Figura 3.5: Captura juego HiddenRoom, vista jugador con visor.

Saturn

La actividad de Saturn (Figuras 3.6) se desarrolla en una tienda de aparatos electrónicos en la que los alumnos deben de realizar una serie de compras de forma colaborativa. Para ello un jugador-cliente tiene varias listas de compra y debe indicarle al jugador-dependiente los productos que tienen que comprar entre los dos. El compañero-dependiente posee un carrito en el que debe ir metiendo los productos que van comprando entre los dos.



Figura 3.6: Captura juego Saturn, vista jugador con lista de la compra.

Coffeeshop

Para la actividad Coffeeshop (Figura 3.7) se ha usado una región virtual en la que se recrea un restaurante con terraza. Es una actividad competitiva por parejas, en la que los jugadores toman el rol de camareros.

En el local y la terraza están sentados diversos clientes que son controlados por el servidor mediante algoritmos de inteligencia artificial (bot). Estos clientes llaman al camarero que maneja el jugador. Este debe desplazarse hasta las cercanías de la mesa donde esté el cliente que lo ha llamado e interactuar con él a través del chat. Debe producirse un diálogo entre el jugador y el cliente, en que el jugador le pregunte al cliente que desea. Después debe acudir a la barra del bar para seleccionar lo que el cliente le ha pedido y volver para servirlo.



Figura 3.7: Captura juego Coffeeshop, jugador hablando con cliente-bot.

Capítulo 4

Análisis del Sistema

En este capítulo se exponen todos los requisitos del sistema. En las distintas secciones del presente capítulo se detallarán los actores del sistema, los requisitos funcionales, los requisitos de información, los requisitos no funcionales, las reglas de negocio.

4.1. Catálogo de actores

Los actores que se tendrán en cuenta a la hora de modelar los casos de uso son los siguientes:

Investigador El docente o investigador será el que utilizará la herramienta para obtener las evidencias de la interacción de los alumnos (o jugadores) con el mundo virtual y entre ellos.

Bases de Datos El actor Bases de Datos (BBDD, Apéndice C) será el sistema del que se obtengan los datos para su posterior procesamiento.

4.2. Requisitos funcionales

El objetivo del presente proyecto es crear un lenguaje de programación, por lo que solo existe un caso de uso principal y bastante básico que se muestra en la Figura 4.1. Las distintas funcionalidades que permite el sistema se encuentran en los tipos de consulta que se pueden realizar, pero siempre siguen la misma interacción con el usuario.

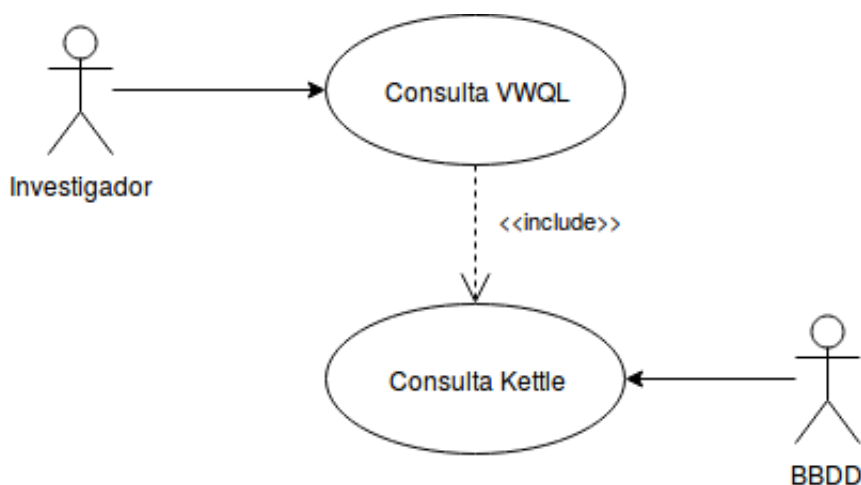


Figura 4.1: Diagrama de caso de uso del sistema.

4.2.1. Caso de Uso: Consulta VWQL

Caso de uso Consulta VWQL

Descripción El usuario realiza una consulta al sistema.

Actores Investigador (Principal), BBDD.

Precondiciones

Postcondiciones El investigador obtiene la información requerida. Se generan el fichero de Transformación Kettle, el fichero con los resultados y los diccionarios se desempaquetan.

Resumen El usuario escribe una consulta en lenguaje VWQL, para obtener información del sistema referente a la interacción de los alumnos en los mundos virtuales.

Escenario principal El usuario obtiene la información requerida a través de un fichero.

Escenario de error El usuario formula incorrectamente la consulta.

Descripción :

Escenario Principal:

1. El usuario escribe una consulta sobre la evidencia de la que desea obtener información y le da a ejecutar.
2. El sistema comprueba si existen los diccionarios.
3. El sistema realiza el análisis léxico, sintáctico y semántico de la consulta.
4. Se genera el fichero de transformación Kettle.
5. Se ejecuta el fichero de transformación de Kettle.
6. Se genera el fichero con los resultados.

Extensiones:

- 2a. Los diccionarios no existen.
 1. El sistema busca y desempaqueta los diccionarios.
- 3a. El análisis detecta un error.
 1. La consulta se para y se muestra un mensaje de error.

Este caso de uso es un ejemplo de súper caso de uso (Figura 4.2), por lo que se podría dividir a su vez en casos de uso hijos, dependiendo del juego que queramos consultar. Cada uno de los casos de uso hijo, se podría sustituir en la descripción del caso de uso padre, es decir, tendría exactamente los mismos pasos, precondiciones, postcondiciones, actores...

En las siguientes subsecciones explicaremos con más detalle que información obtenemos de los indicadores correspondientes a cada consulta.

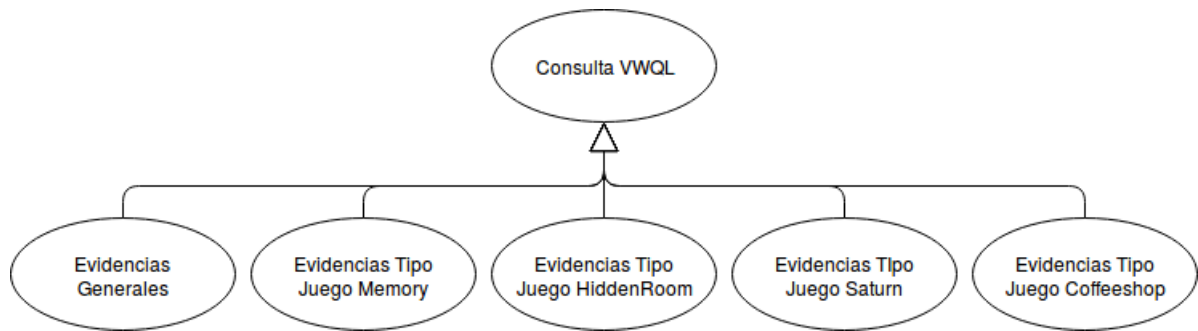


Figura 4.2: Diagrama de super caso de uso del sistema.

4.2.2. Caso de uso: Evidencias generales

El caso de uso *Evidencias generales* a su vez se divide en varias consultas como se muestra en la Figura 4.3. En las subsubsecciones del presente apartado se describirá que hace cada una de ellas.

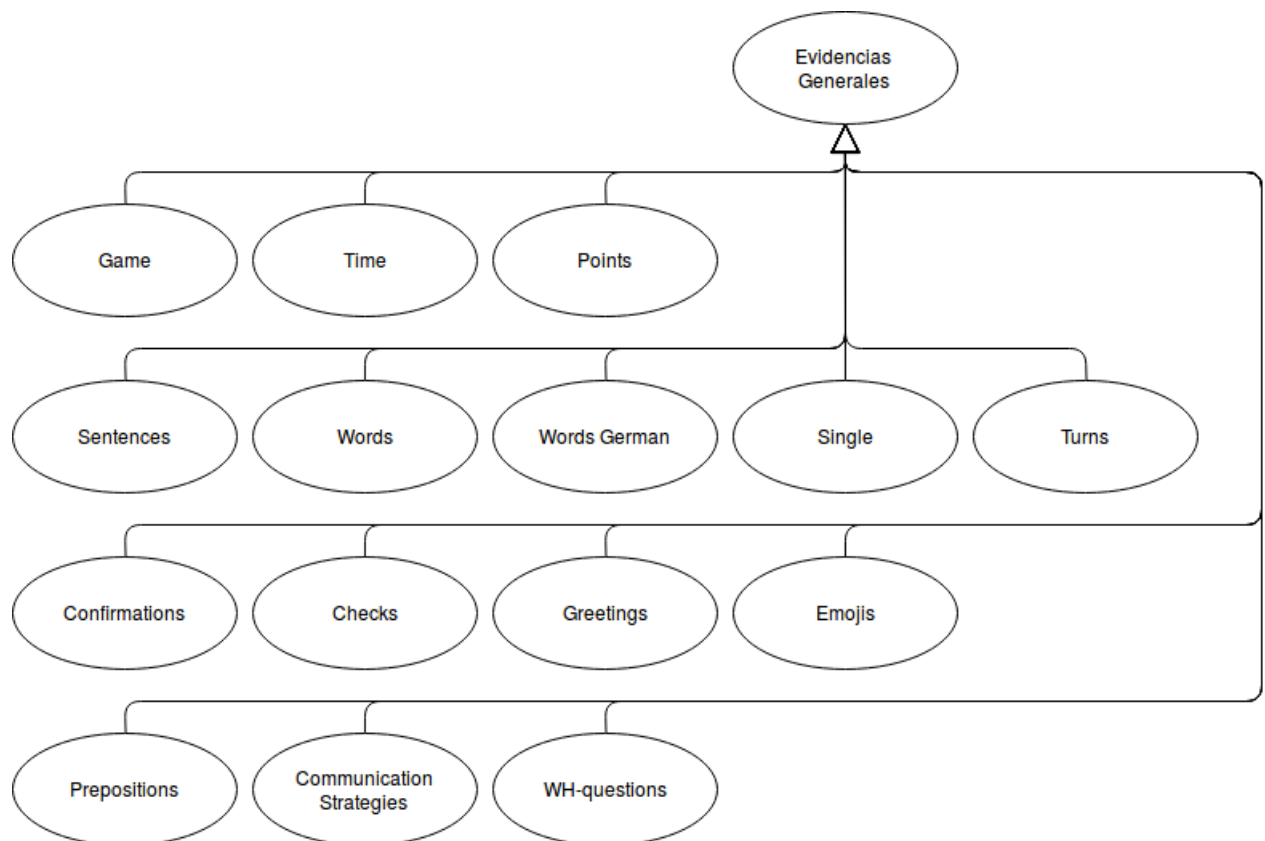


Figura 4.3: Diagrama de caso de uso *Evidencias generales* del sistema.

Game

Descripción Esta consulta extrae información con todo lo referente a las partidas jugadas: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida, duración de la partida y puntuación.

Time

Descripción Esta consulta extrae información con todo lo referente al tiempo de las partidas jugadas: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida y duración de la partida.

Points

Descripción Esta consulta extrae información con todo lo referente a la puntuación de las partidas jugadas: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida y puntuación obtenida.

Sentence

Descripción Esta consulta extrae todos los mensajes enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida y mensaje.

Words

Descripción Esta consulta extrae todas las palabras de los mensajes enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, palabra y mensaje.

Words german

Descripción Esta consulta extrae todas las palabras en alemán de los mensajes enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, palabra y mensaje.

Single

Descripción Esta consulta extrae todos los mensajes, de una sola palabra, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Turns

Descripción Esta consulta extrae todos los mensajes enviados por el chat durante la partida, y contabiliza los turnos de palabra de la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, turno y mensaje.

Confirmations

Descripción Esta consulta extrae todos los mensajes de confirmación de que algo es correcto, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Checks

Descripción Esta consulta extrae todos los mensajes, que comprueban si se presta atención o se entiende, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Greetings

Descripción Esta consulta extrae todos los mensajes, en los que se saluda o se despide, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Emojis

Descripción Esta consulta extrae todos los mensajes, que contienen emoticonos, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Prepositions

Descripción Esta consulta extrae todos los mensajes, en los que se usan preposiciones, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

WH-questions

Descripción Esta consulta extrae todos los mensajes, que contienen preguntas informativas, enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

Communication Strategies

Descripción Esta consulta extrae todos los mensajes, que contienen ciertas estrategias de comunicación (mantener la atención por ejemplo), enviados por el chat durante la partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, y mensaje.

4.2.3. Caso de Uso: Evidencias Memory

El caso de uso *Evidencias Memory* se divide en tan solo dos consultas, las cuales se muestran en la Figura 4.4. Las dos consultas se describen a continuación.

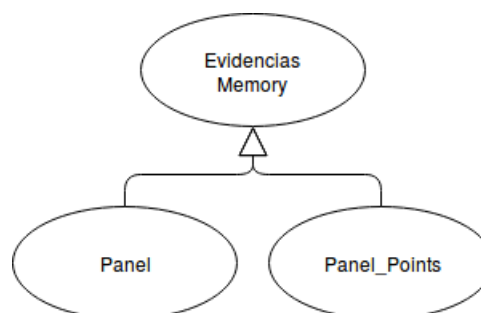


Figura 4.4: Diagrama de caso de uso *Evidencias Memory* del sistema.

Panel

Descripción Esta consulta extrae las veces que se ha jugado a un panel y los puntos que se ha obtenido cada vez. Los datos que se muestran son: nombre del juego, nombre del jugador, número del panel, veces que se jugó, puntos totales, y lista de puntos.

Panel_Points

Descripción Esta consulta extrae las veces que se ha jugado a un panel y los puntos que se ha obtenido cada vez. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora del mensaje, tiempo de juego, numero del panel y puntuación.

4.2.4. Caso de Uso: Evidencias HiddenRoom

El caso de uso *Evidencias HiddenRoom* solo se divide en una consulta. Esta consulta se muestra en la Figura 4.5 y se describe a continuación.

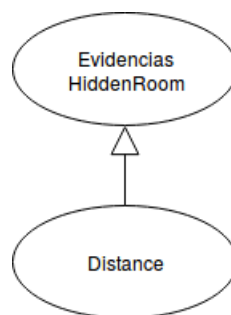


Figura 4.5: Diagrama de caso de uso *Evidencias HiddenRoom* del sistema.

Distance

Descripción Esta consulta extrae la distancia que ha recorrido un jugador en el mundo virtual durante una partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida, y distancia recorrida en metros (virtuales).

4.2.5. Caso de Uso: Evidencias Coffeeshop

El caso de uso *Evidencias Coffeeshop*, se divide a su vez en dos consultas. Estas consultas se muestran en la Figura 4.6 y se describen a continuación.

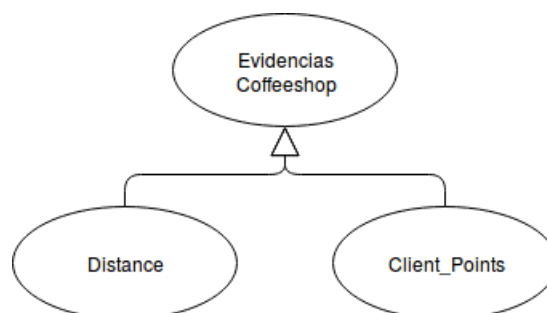


Figura 4.6: Diagrama de caso de uso *Evidencias Coffeeshop* del sistema.

Distance

Descripción Esta consulta extrae la distancia que ha recorrido un jugador en el mundo virtual durante una partida. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida, y distancia recorrida en metros (virtuales).

Client_Points

Descripción Esta consulta extrae todas las veces que un jugador ha servido a un cliente, y la puntuación que ha obtenido. Los datos que se muestran son: nombre del juego, nombre del jugador, fecha y hora de finalización de la partida, puntos totales en la partida, cliente al que sirvió y puntos obtenidos con el cliente.

4.3. Requisitos de información

EvalSim tiene como objetivo extraer información de la interacción que tienen los jugadores en los mundos virtuales y poder evaluarla. El sistema está pensado para el despliegue, de estos mundos virtuales, que actualmente tiene en la Universidad de Cádiz. Por tanto, para saber como extraer el flujo de datos de inicio, hay que conocer la tecnología actual y los datos de los que disponemos. Estos datos posteriormente serán procesados de forma que obtengamos información de la interacción que se produce en los mundos virtuales.

Lo primero que debemos analizar son las bases de datos de OpenSim, las bases de datos de los juegos y como se conectan. Para ver la interconexión entre las diferentes bases de datos tenemos la Figura 4.7 en la Subsección 4.3.1, y para conocer más a fondo las bases de datos y que contiene cada una de ellas dispondremos del Apéndice C.

Una vez analizados los datos y sabiendo que contiene cada una de las tablas de las bases de datos, necesitamos realizar un mapeo entre los datos que queremos obtener de las bases de datos y las consultas que vamos a realizar en Kettle para obtener los indicadores.

4.3.1. Mundos Virtuales

La plataforma de mundos virtuales desplegada en la Universidad de Cádiz, cuenta con varias bases de datos alojadas en servidores. Todas las bases de datos de las que disponemos están detalladas en el Apéndice C.

EvalSim solo necesita una parte de los datos que se almacenan en ellas y también saber como se enlazan sus registros dentro de la base de datos. De forma global, en la Figura 4.7, podemos ver como se interconectan las bases de datos. En los siguientes puntos describiremos la estructura con más detalle.

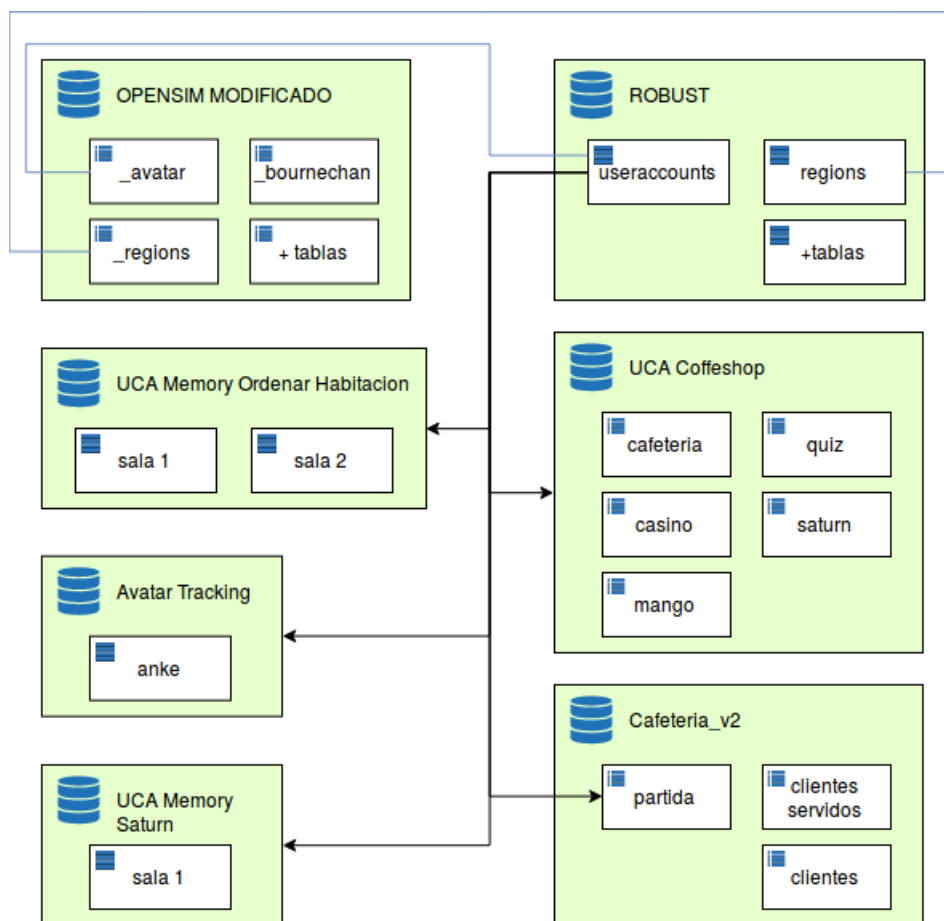


Figura 4.7: Diagrama de bases de datos de los mundos virtuales.

ROBUST

La base de datos ROBUST (Apéndice C.7) es una base de datos generada automáticamente por OpenSim al crear el grid de mundos virtuales. Es la base de datos enlace entre todas las regiones de los mundos virtuales y los usuarios. En la Figura 4.7 podemos ver que absolutamente todas las bases de datos se conectan con ROBUST, en concreto con la tabla *useraccounts*.

En esta base de datos solo hay dos tablas que nos interesa conocer, que son *useraccounts* y *regions* (Apéndices C.7.11 y C.7.9). De *regions* podemos extraer datos de identificación y localización de las diferentes regiones que existen dentro del mundo virtual. De *useraccounts* podemos obtener todos los datos correspondientes a identificación y datos personales de usuario. Aunque internamente OpenSim use un *Universally Unique Identifier (UUID)* que corresponde con el campo *PrincipalID* de la tabla *useraccounts*, en los juegos se usa como identificador el nombre y apellido del avatar (campos *FirstName* y *LastName*). Debido a problemas técnicos, se decidió usar el nombre y apellidos como identificador en las bases de datos de los juegos.

OPENSIM MODIFICADO

Las bases de datos OPENSIM MODIFICADO son también generadas automáticamente por OpenSim, solo que sirven para administrar una o varias regiones. Estas bases de datos contienen todo lo que

existe dentro de cada región o regiones, es decir, contiene la forma del terreno, objetos 3D, texturas, audios... Cada una de estas bases de datos es formada por las tablas de OPENSIM y OPENSIM MODIFICADO (Apéndices C.8 y C.9). Como se explica en los apéndices, OPENSIM MODIFICADO es un añadido a la base de datos que genera automáticamente OpenSim.

Para el proyecto, se ha generado una sola base de datos unificada que contiene todas las tablas de OPENSIM MODIFICADO (Apéndice C.9) y se ha hecho un volcado de cada una de las bases de datos que administran las regiones.

Dentro de OPENSIM MODIFICADO, todas las tablas contienen datos útiles para analizar en investigación. En este proyecto solo nos hemos centrado en analizar el chat, que corresponde con la tabla *bournechannel* (Apéndice C.9.3). Del chat se obtienen bastantes indicadores, como se muestra en la subsección 4.2.2.

En la Figura 4.7, podemos observar que también se han resaltado otras dos tablas, que son *_avatar* y *_regions*. De ambas tablas podemos destacar en la figura que existe una conexión directa y no direccional con las tablas *useraccounts* y *regions* de la base de datos ROBUST. Esto se ha hecho porque se genera una versión reducida de estas dos tablas desde ROBUST. Esta modificación se hizo a la versión por defecto de OpenSim para ser más fácilmente analizada por los investigadores, y de esta forma no tener que estar consultando en ROBUST estos datos.

Avatar Tracking

La base de datos Avatar Tracking (Apéndice C.1) contiene registros de posicionamiento de los avatares. A través del cruce entre esta base de datos y las que contienen los datos de los juegos, podemos saber cual fue el recorrido y la distancia que recorrió cada uno de los jugadores durante las partidas.

UCA Memory Ordenar Habitacion

Esta base de datos (Apéndice C.3) contiene los datos de los juegos *MemoryHiddenRoom* en la tabla *sala_1* y *HiddenRoom* en la tabla *sala_2* que se pueden ver en la Figura 4.7.

En la tabla *sala_1* hay datos sobre quien jugó, la fecha y hora en que jugó, los puntos que obtuvo, el panel que hizo y el tiempo que tardó.

En la tabla *sala_2* hay datos sobre la pareja que jugó junto con el rol de cada miembro, los puntos que obtuvieron en conjunto, la fecha y hora en que jugó, y el tiempo que tardaron.

UCA Coffeshop

Esta base de datos (Apéndice C.4) contiene 5 tablas con datos sobre 5 juegos que se desarrollaron en la Universidad Autónoma de Madrid. Sin embargo, en la Universidad de Cádiz solo se ha hecho uso del juego *Saturn*, por eso es el único juego que se ha incluido en el DSL. En trabajos futuros se pueden incluir los demás juegos.

El juego *Saturn* también es de rol y se realizaba por parejas, pero a diferencia del juego *HiddenRoom*, se almacena cada miembro por separado junto con su rol. Por tanto, de esta tabla podemos obtener nombre del jugador, rol, fecha y hora de finalización de la partida, puntos obtenidos de forma conjunta y tiempo que tardó la pareja en realizar la actividad.

UCA Memory Saturn

La base de datos UCA Memory Saturn (Apéndice C.5), contiene una tabla denominada *sala_1* que almacena datos sobre el juego Memory con vocabulario de Saturn. De esta tabla podemos sacar los mismos datos que en MemoryHiddenRoom: quien jugó, la fecha y hora en que jugó, los puntos que obtuvo, el panel que hizo y el tiempo que tardó.

Cafeteria v2

Esta base de datos (Apéndice C.2) contiene información sobre una variante del juego *cafeteria* que se almacena en la base de datos UCA Coffeshop (Subsubsección 4.3.1). Como se puede observar en la Figura 4.7, la tabla *useraccounts* de ROBUST se enlaza directamente con la tabla *partida*. Esto se debe a que, a diferencia de los demás casos, solo hace referencia a los usuarios en esta tabla, y no en todas las de la base de datos.

Este juego de rol se realizaba por parejas, pero de forma competitiva, por lo que la puntuación es individual. También se almacenan los puntos por cliente servido en una partida.

De la tabla *partida* podemos obtener los nombres de los jugadores, los puntos individuales que obtuvieron, el tiempo que duró la partida, y la fecha y hora de finalización.

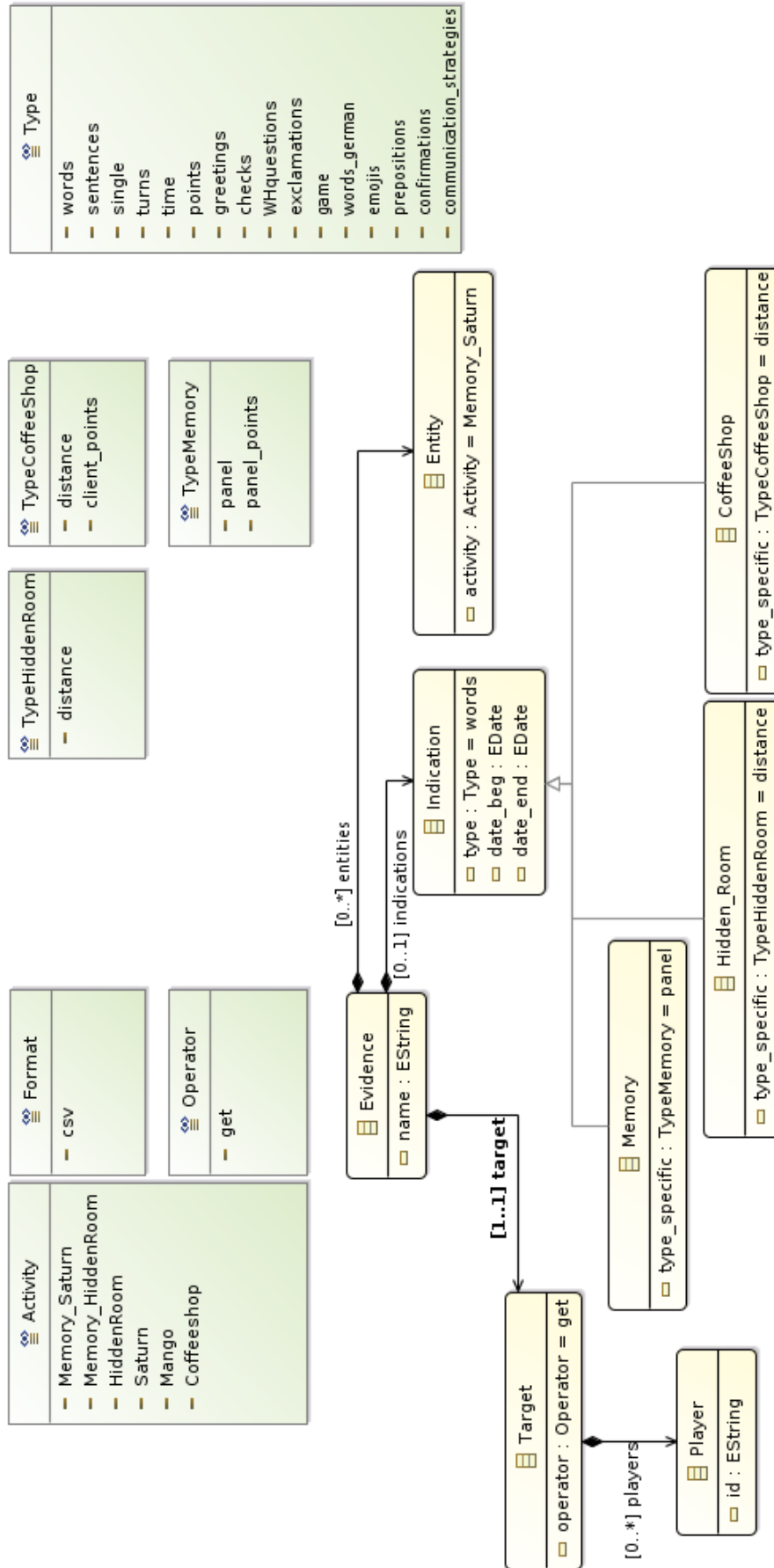
De la tabla *clientes_servidos* podemos obtener el número de clientes que sirvió un jugador en una partida, y los puntos que obtuvo de dicho cliente.

4.3.2. Metamodelo

Los DSL son herramientas para abordar problemas concretos. Para desarrollar un DSL, el primer paso es diseñar el metamodelo. El metamodelo es el conjunto de conceptos del dominio a modelar (metaclases) y las relaciones entre ellos (metasociaciones). Para realizar el metamodelo necesitamos un análisis previo con el que poder determinar el dominio concreto del problema, junto con sus reglas y restricciones.

El metamodelo de EvalSim (Figura 4.8) define el dominio del estudio de los mundos virtuales. En este metamodelo aparecen definidos los conceptos y relaciones que existen, y que serán detallados a continuación:

- **Evidence:** es el concepto principal que permite identificar cada una de las consultas que vayamos a evaluar. Es la raíz del metamodelo, ya que las demás metaclases son composiciones de Evidence. Evidence contiene un atributo llamado *name*, que sirve para dar nombre a la consulta y a los ficheros de salida que el DSL generará.
- **Target:** este concepto representa el objetivo que queremos extraer de los estudiantes. Solo hemos incluido en el DSL un operador que es *get*, pero queda abierto para ser ampliado con más operadores.
- **Player:** representa a los jugadores que queremos evaluar en la consulta. Es un parámetro opcional ya que podemos seleccionar jugadores de forma concreta o no introducir ninguno y evaluarlos todos.
- **Entity:** representa cada uno de los juegos que pretendemos evaluar en la consulta. Es un parámetro opcional ya que podemos seleccionar juegos de forma concreta o evaluarlos todos si no introduci-



29
Figura 4.8: Metamodelo del EvalSim.

mos ninguno. Hay que tener en cuenta que las entidades que seleccionemos deben concordar con el indicador escogido, aunque no se refleje en el metamodelo.

- **Indication:** representa el tipo de información queremos consultar. Dependiendo de los atributos *type* o *type_specific* podemos seleccionar dicha información. También se puede seleccionar un rango de fechas para realizar la consulta.

Como podemos observar en la Figura 4.8, existe una especialización de esta metaclass. Se debe a que existen indicios que única y exclusivamente se pueden aplicar a un tipo de juego concreto.

- *Generales:* representa las consultas básicas que se pueden realizar en los juegos, y se usa al seleccionar cualquier indicador de la lista de *Type*.
- *Memory:* representa las consultas que se pueden realizar de los juegos tipo Memory, y se usa al seleccionar un indicador de la lista *TypeMemory*
- *HiddenRoom:* representa las consultas que se pueden realizar del juego HiddenRoom, y se usa al seleccionar un indicador de la lista *TypeHiddenRoom*
- *Coffeeshop:* representa las consultas que se pueden realizar del juego Coffeeshop, y se usa al seleccionar un indicador de la lista *TypeCoffeeshop*

Como hemos comentado anteriormente, el atributo *type* o *type_specific* seleccionado, debe ser compatible con las entidades seleccionadas. En caso de que no se seleccione ninguna entidad pero si se seleccione un indicio específico de un juego, automáticamente el sistema evaluará las entidades compatibles con la consulta. Esto se verá en la sección Implementación.

4.4. Requisitos no funcionales

Los requisitos no funcionales de un sistema son requisitos que imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener y que están relacionadas con la calidad del software.

El sistema EvalSim debe satisfacer los siguientes requisitos:

- **Portabilidad:** la herramienta debe ser compatible con las diferentes plataformas de PC. El software ha sido desarrollado como plugin de Eclipse, por tanto en cualquier sistema que pueda ser instalado Eclipse, podría ser usado. Esto quiere decir que EvalSim puede ser usado en los sistemas más extendidos como son Linux, Windows y Mac OS.
- **Extensibilidad:** el software ha sido diseñado lo más genérico posible para que se pueda ampliar con nuevos juegos dentro de los mundos virtuales y con nuevos indicadores.
- **Usabilidad:** como ya se ha comentado a lo largo del proyecto, este es uno de los puntos principales. El software se ha diseñado lo más simple e intuitivo posible para que pueda ser usado por cualquier investigador, tenga o no conocimientos informáticos previos.
- **Seguridad:** el sistema se conectará al servidor de la plataforma OpenSim. EvalSim tendrá acceso al backup modificado que se ha creado, y solo se puede consultar el contenido.
- **Fiabilidad:** los juegos en los mundos virtuales generan gran cantidad de datos en la interacción de los jugadores con el entorno y con otros jugadores. Es por ello que al manejar estos datos EvalSim, se esperan resultados que representan la realidad, sin errores y que proporcione información fiable. Para que la información que se vaya a obtener sea fiable, se ha realizado un análisis exhaustivo de

las diferentes bases de datos que soportan los mundos virtuales, y se ha tenido especial cuidado en que al procesar los datos nada falle.

- **Mantenibilidad:** se ha diseñado y desarrollado un software, que permita seguir siendo desarrollado por terceras personas ajenas al proyecto inicial, sin que ocasione a las mismas un gran esfuerzo o una gran inversión de tiempo.
- **Comunicación con sistemas externos:** la conexión y consulta a las bases de datos se deben realizar desde un equipo con los permisos necesarios. Este requisito no entra dentro del ámbito del proyecto. La conexión de EvalSim con las bases de datos se realiza a través de la transformación Kettle. Se ha creado un perfil de solo consulta a las bases de datos que utiliza el sistema, y que será los credenciales que se utilicen en las consultas.
- **Privacidad:** el sistema cumple con la normativa y legislación vigentes en materia de protección de datos, siendo de aplicación la Ley Orgánica 15/1999 del 13 de diciembre de Protección de Datos de Carácter Personal. Todos los datos generados en los mundos virtuales y almacenados en las bases de datos, se realizan de forma completamente anónima y con el consentimiento de los jugadores de prueba. Los datos son generados durante pruebas controladas en un entorno instructivo como actividad de aprendizaje complementario.

4.5. Estudio de alternativas tecnológicas

Hasta el momento, la alternativa tecnológica a EvalSim, y que es la que se ha estado usando, es realizar el proceso de forma manual.

Un investigador que necesita analizar una serie de datos, pide a un técnico que los recolecte de las bases de datos a través de consulta SQL, y posteriormente procesar los resultados con programas o analizarlos directamente uno por uno. Es un proceso costoso y lento. Aparte de esto, mientras se sigan usando los mundos virtuales, se seguirá aumentando el volumen de datos y cada vez será mas complejo realizar el proceso de forma manual.

Por tanto, actualmente no existe ningún sistema automatizado que proporcione todos los indicadores requeridos de la plataforma de mundos virtuales desarrollada en la Universidad de Cádiz.

Capítulo 5

Diseño del Sistema

En este capítulo se procede a definir el sistema con todo detalle. Se recoge la arquitectura general del sistema de información y el diseño de componentes software.

5.1. Diseño de la arquitectura

En la Figura 5.3 podemos ver el flujo de diseño, y las etapas que se han seguido. El flujo que se observa en la figura coincide con los pasos necesarios para el desarrollo de un DSL mediante Ingeniería Dirigida por Modelos (MDE).

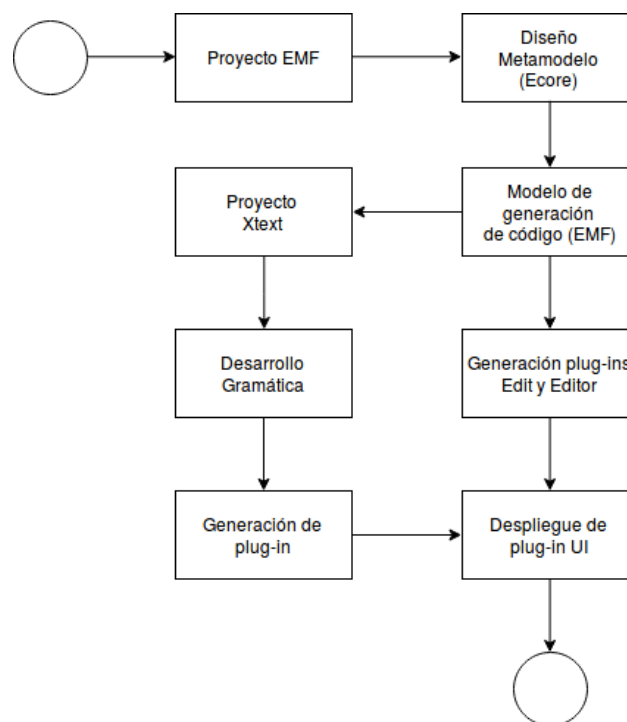


Figura 5.1: Diseño de la arquitectura lógica.

5.2. Arquitectura física

El sistema completo, incluyendo como se generan los datos en las bases de datos de OpenSim, se constituye por todos los componentes hardware que aparecen en la Figura 5.2. Los elementos hardware se detallan a continuación:

- **Servidor OpenSim:** Dentro del servidor se encuentran las bases de datos de los mundos virtuales y de los juegos, y las aplicaciones servidor de OpenSim para que los alumnos puedan conectarse a los mundos virtuales.
- **Ordenador de usuario de EvalSim:** El equipo es el que el investigador o docente vaya a utilizar EvalSim.
- **Ordenador de usuario de OpenSim:** El equipo en el que el alumno vaya a utilizar los mundos virtuales.

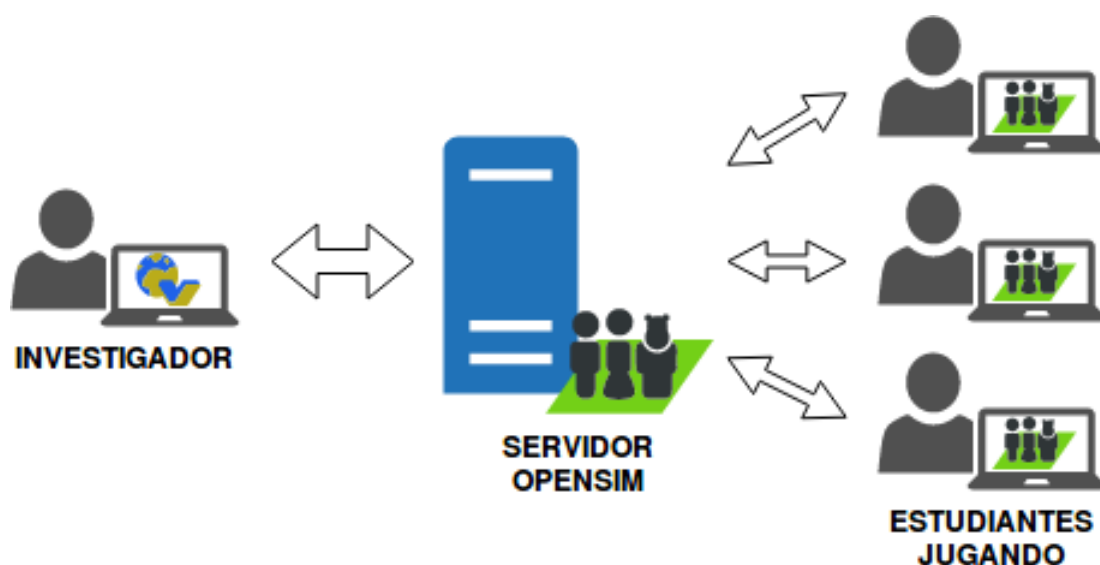


Figura 5.2: Diagrama de la arquitectura física actual de EvalSim.

Durante el proyecto no se va a utilizar el servidor remoto, sino que se hará todo de forma local, con copias de seguridad modificadas de las bases de datos del servidor.

5.3. Arquitectura lógica

En el diagrama de la Figura 5.3 se puede ver la disposición de los elementos que componen la arquitectura lógica de EvalSim.

A continuación pasamos a definir brevemente los diferentes componentes software que componen el sistema:

- **Eclipse:** es el IDE en el que se instalará y ejecutará la aplicación. Dentro de Eclipse haremos uso de los componentes Ecore, Xtext y Xtend. Estos componentes quedan reflejados en la Figura 5.3 con el elemento *ANALIZADOR/GENERADOR*, y el IDE Eclipse correspondería a todo el área azul, al igual que EvalSim.

- **EvalSim:** es el software desarrollado en este proyecto. Se instala en el IDE Eclipse como plugin. En la Figura 5.3 correspondería con el área azul.
- **OpenSim:** aplicación servidor de mundos virtuales al que se conectarán los alumnos para jugar. No está representado en la Figura 5.3, ya que no forma parte de EvalSim como tal, pero se conectaría con el elemento *BBDD MUNDOS VIRTUALES*.
- **MySQL:** sistema de gestión de bases de datos relacional usado para almacenar todos los datos de los mundos virtuales OpenSim y de los juegos desarrollados dentro de éstos. En la Figura 5.3 corresponde con el elemento *BBDD MUNDOS VIRTUALES*.
- **Pentaho:** conjunto de programas libres que permiten generar inteligencia empresarial. Pentaho incluye herramientas para generar informes, minería de datos, ETL... Dentro de todos estos componentes, el que usaremos será Kettle que es una herramienta ETL (Extracción, Transformación y Carga). En la Figura 5.3 corresponde al elemento *KETTLE*.

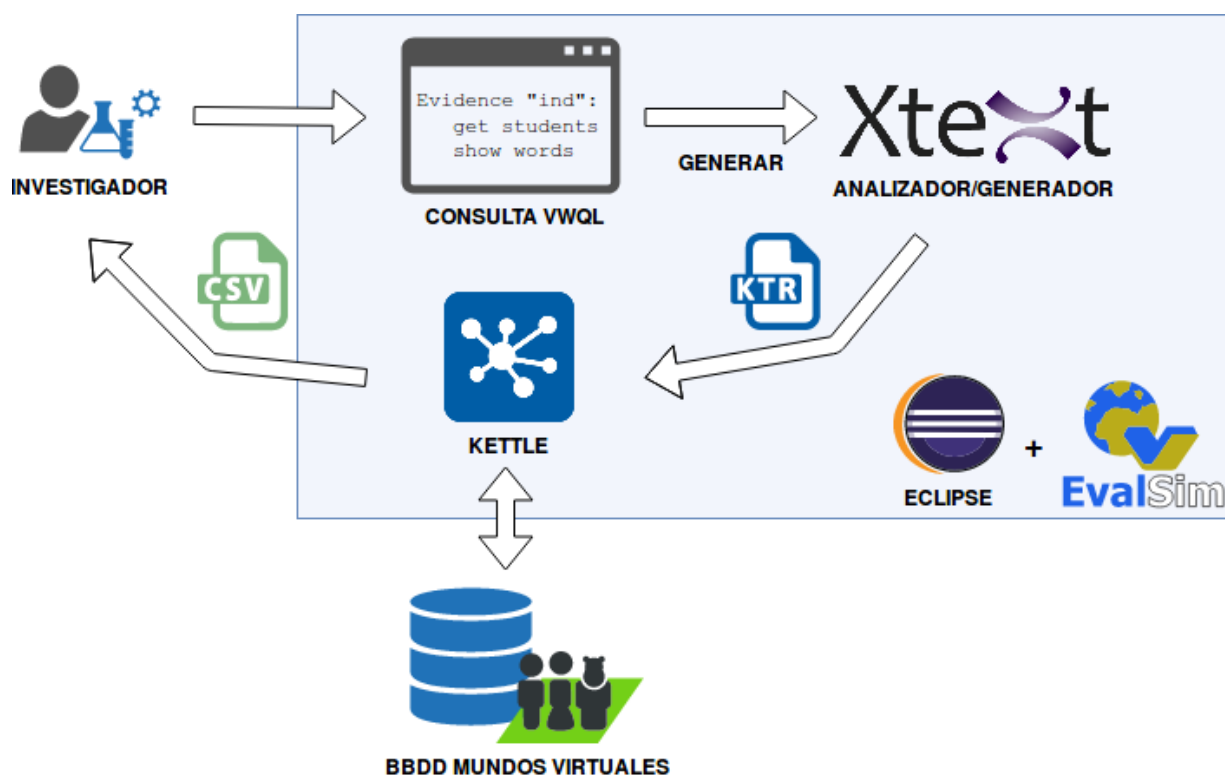


Figura 5.3: Diagrama de la arquitectura lógica de EvalSim.

Tanto los componentes de Eclipse como Kettle de Pentaho serán descritos con detalle en Implementación en la Sección 6.5.

5.3.1. Transformaciones Kettle

Como se ha explicado en la subsección anterior, las transformaciones son los pasos que Kettle ejecuta para realizar un proceso ETL. En este apartado describiremos los pasos de PDI que se van a usar en las transformaciones y se mostrará como debe ser cada fichero de transformación según la consulta.

Todas las transformaciones que incluya procesamiento del chat, solo se aplica a los juegos HiddenRoom, Saturn y Coffeeshop. También comentar que en éstos casos los datos de entrada del juego Saturn necesitan ser normalizados y por ello cuenta con bastantes pasos más que no se van a incluir en el proceso estándar.

Hay consultas en las que se genera un identificador con el elemento *Sequence* de Kettle. Este identificador, al ser generado, no puede ser usado para identificar una partida entre consultas distintas, o entre resultados de ejecuciones distintas aun siendo la misma consulta, solo sirve como referencia dentro de los datos de salida de una consulta.

Los datos del chat de la base de datos OpenSim, en realidad contiene todos los datos de todos los chat de los mundos virtuales. Esta base de datos se ha generado al unificar todas las bases de datos de las regiones virtuales en una sola. El chat es el más importante para este proyecto pero también se ha hecho lo mismo con varias tablas más y se han eliminado otras que carecen de importancia para este proyecto o para proyectos futuros.

Transformación GAME

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Constant*: se añade el nombre del juego al que pertenece.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sort Rows* se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Unique Rows*: se unen las filas de todos los juegos, eliminando las repetidas.
- *Sort Rows*: se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación TIME

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Constant*: se añade el nombre del juego al que pertenece.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sort Rows* se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Unique Rows*: se unen las filas de todos los juegos, eliminando las repetidas.
- *Sort Rows*: se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación POINTS

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Constant*: se añade el nombre del juego al que pertenece.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sort Rows*: se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Unique Rows*: se unen las filas de todos los juegos, eliminando las repetidas.
- *Sort Rows*: se ordenan las filas en función de los campos clave especificados en orden ascendente.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación TURNS

Se contabiliza un turno cada vez que un jugador distinto mande un mensaje al chat. Por ejemplo, el jugador 1, manda 3 mensajes al chat (turno 1), el jugador 2 manda 1 mensaje al chat (turno 2), y el jugador 1 vuelve a mandar 2 mensajes (turno 3).

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Calculator*: calcula la fecha y hora de inicio y de fin de la partida.
- *Sequence*: genera un identificador para cada partida.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Table Input*: extrae los datos del chat de la base de datos OpenSim.
- *Join Rows*: se realiza una unión condicional entre el flujo de datos de cada juego y el flujo de datos del chat. La condición es que las filas del chat estén entre la fecha de inicio y la fecha de fin de la partida. Así localizamos lo que se ha hablado durante una partida.
- *Sort Rows*: se ordenan las filas de cada juego en función de los campos clave especificados en orden ascendente.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sorted Merge*: se unen todas las filas de todos los juegos al mismo flujo de datos de forma ordenada. El orden es ascendente en función de los campos clave seleccionados.
- *Analytic Query*: compara la fila actual con la fila anterior en el flujo de datos comprobando el *id_game* y el nombre del jugador, generando un campo nuevo por cada comparación.
- *Script Value*: paso personalizado que nos permite contabilizar el número de turnos comprobando los campos, *id_game* con *id_game* de la fila anterior, nombre con nombre de fila anterior.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación DISTANCE

Se contabiliza la distancia que ha recorrido un jugador durante una partida. Para ello se recogen todos los registros de la base de datos Avatar Tracking y se cruzan con los de las partidas de los juegos HiddenRoom y Coffeshop. El cruce de datos se realiza comparando que los registros del posicionamiento de los jugadores estén entre la fecha inicio y la fecha fin de la partida.

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Calculator*: calcula la fecha y hora de inicio y de fin de la partida.
- *Sequence*: genera un identificador para cada partida.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Table Input*: extrae los datos de posicionamiento de la base de datos Avatar Tracking.
- *Join Rows*: se realiza una unión condicional entre el flujo de datos de cada juego y el flujo de datos del posicionamiento. La condición es que las filas de posicionamiento estén entre la fecha de inicio y la fecha de fin de la partida. Así localizamos puntos por los que ha pasado el jugador durante la partida.
- *Sort Rows*: se ordenan las filas de cada juego en función de los campos clave especificados en orden ascendente.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sorted Merge*: se unen todas las filas de todos los juegos al mismo flujo de datos de forma ordenada. El orden es ascendente en función de los campos clave seleccionados.
- *Analytic Query*: compara la fila actual con la fila anterior en el flujo de datos comprobando el *id_game*, el nombre del jugador, la coordenada X y la coordenada Y; generando un campo nuevo por cada comparación.
- *Script Value*: paso personalizado que nos permite contabilizar la distancia recorrida comparando los campos: *id_game* con *id_game* de la fila anterior si son iguales; nombre con nombre de fila anterior si son iguales; coordenada X con coordenada X de la fila anterior si son distintos; y coordenada Y con coordenada Y de la fila anterior si son distintos. Si se cumple la condición, se calcula la distancia en línea recta entre las dos posiciones y se suma a la distancia total recorrida.
- *Group By*: se realiza la operación de agrupar todas las filas de una partida y se limita a la mayor distancia.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformaciones con chat

En todas las transformaciones con el chat, se realizan previamente estos pasos:

- *Table Input*: extrae los datos de las bases de datos de los juegos seleccionados.
- *Calculator*: calcula la fecha y hora de inicio y de fin de la partida.
- *Sequence*: genera un identificador para cada partida.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Table Input*: extrae los datos del chat de la base de datos OpenSim.
- *Join Rows*: se realiza una unión condicional entre el flujo de datos de cada juego y el flujo de datos del chat. La condición es que las filas del chat estén entre la fecha de inicio y la fecha de fin de la partida. Así localizamos lo que se ha hablado durante una partida.
- *Sort Rows*: se ordenan las filas de cada juego en función de los campos clave especificados en orden ascendente.
- *Select Values*: se seleccionan los campos de salida en el orden especificado y el resto se elimina del flujo
- *Sorted Merge*: se unen todas las filas de todos los juegos al mismo flujo de datos de forma ordenada. El orden es ascendente en función de los campos clave seleccionados.

Transformación SENTENCES

- Pasos descritos en la subsubsección 5.3.1.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación SINGLE

- Pasos descritos en la subsubsección 5.3.1.
- *Replace in String*: se eliminan los signos de puntuación de cada campo que contiene el mensaje.
- *Janino*: añade dos campos a cada fila del flujo de datos. En el primer campo cuenta el número de palabras y el segundo cuenta el número de letras.
- *Filter Rows*: se eliminan del flujo de datos las filas que no cumplan que el número de palabras sea uno y el número de letras mayor que cero. La segunda comprobación se realiza porque con solo la primera condición se contabilizan campos no vacíos no nulos, como por ejemplo saltos de línea.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación WORDS

- Pasos descritos en la subsubsección 5.3.1.
- *Split Field to Rows*: se divide cada mensaje del chat en palabras, y se crea una fila por palabra.
- *Replace String*: se eliminan los signos de puntuación de cada campo que contiene las palabras.
- *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras.
- *Filter Rows*: se eliminan las filas que contengan el campo palabra nulo.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación WORDS GERMAN

- Pasos descritos en la subsubsección 5.3.1.
- *Split Field to Rows*: se divide cada mensaje del chat en palabras, y se crea una fila por palabra.
- *Replace in String*: se eliminan los signos de puntuación de cada campo que contiene las palabras.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras.
- En paralelo se ejecuta:
 - *Text File Input*: se extraen los datos del diccionario de alemán.
 - *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras del flujo de datos del diccionario.
- *Stream Lookup*: se cruzan los datos del flujo de los juegos con el flujo de datos del diccionario, comparando si coinciden el campo que contienen las palabras del diccionario con el campo que contiene las palabras del chat.
- *Filter Rows*: se eliminan las filas que contengan el campo palabra nulo.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación EMOJIS

- Pasos descritos en la subsubsección 5.3.1.
- *Split Field to Rows*: se divide cada mensaje del chat en palabras, y se crea una fila por palabra.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras.

- En paralelo se ejecuta:
 - *Text File Input*: se extraen los datos del diccionario de emojis.
 - *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras del flujo de datos del diccionario.
- *Stream Lookup*: se cruzan los datos del flujo de los juegos con el flujo de datos del diccionario, comparando si coinciden el campo que contienen las palabras del diccionario con el campo que contiene las palabras del chat.
- *Filter Rows*: se eliminan las filas que contengan el campo palabra nulo.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Transformación PREPOSITIONS

Esta transformación permite obtener mensajes que contengan el uso de preposiciones. Es un indicador muy importante ya que el juego HiddenRoom se creó, en parte, para reforzar el uso de estas estructuras gramaticales.

Para realizar esta transformación, previamente se han preparado unos diccionarios con preposiciones y con nombres, cada uno de ellos con el género (masculino, femenino o neutro). Estos diccionarios se han generado para que al concatenar una preposición con un nombre, formen una expresión regular. De esta forma al cruzar los datos con los mensajes del chat, podemos obtener los mensajes que contienen estas estructuras.

- Pasos descritos en la subsubsección 5.3.1.
- *Replace in String*: se eliminan los signos de puntuación de cada campo que contiene los mensajes.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras.
- En paralelo se ejecuta:
 - *Text File Input*: de forma paralela se extraen los datos del diccionario de preposiciones, y si es masculino, femenino o neutro.
 - *Text File Input*: de forma paralela se extraen los datos del diccionario de nombres, y si es masculino, femenino o neutro.
 - *Join Rows*: se realiza el producto cartesiano de los flujos de datos de las preposiciones y de los nombres, comparando que si la preposición es masculino se une con los nombres masculinos o neutros, y si la preposición es femenina se une con los nombres femeninos.
 - *Calculator*: concatenamos en cada fila el campo de preposición con el campo del nombre.
 - *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las expresiones regulares.
- *Join Rows*: se cruzan los datos del flujo de los juegos con el flujo de datos de los diccionarios, comparando si coinciden el campo que contienen la expresión regular con el campo que contiene los mensajes del chat.

- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

Otras transformaciones chat

Todas las siguientes consultas se ejecutan exactamente de la misma forma: GREETINGS, WQ-QUESTIONS, CONFIRMATIONS, CHECKS, COMMUNICATION STRATEGIES. La única diferencia entre ellas es que se usan diferentes diccionarios con expresiones regulares. Cada expresión tiene su propia expresión regular, por eso no podemos realizarlo de forma dinámica, concatenándole una expresión predeterminada por delante y por detrás de la palabra del diccionario.

- Pasos descritos en la subsubsección 5.3.1.
- *Replace in String*: se eliminan los signos de puntuación de cada campo que contiene los mensajes.
- *Replace in String*: se reemplaza el nombre de la región de la base de datos por el nombre del juego.
- *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las palabras.
- En paralelo se ejecuta:
 - *Text File Input*: se extraen los datos del diccionario de la consulta seleccionada.
 - *String Operations*: se ponen en minúscula todos los caracteres del campo que contiene las expresiones regulares.
- *Join Rows*: se cruzan los datos del flujo de los juegos con el flujo de datos de los diccionarios, comparando si coinciden el campo que contienen la expresión regular con el campo que contiene los mensajes del chat.
- *Text File Output*: se seleccionan los datos que van a ser devueltos en el fichero CSV y se genera el archivo.

5.3.2. Estructura fichero Kettle

En esta subsección pasaremos a explicar la estructura que tiene un fichero de transformación con extensión *.ktr* de PDI. El fichero está escrito en XML, por lo que la interpretación es bastante sencilla. Para generar los procesos ETL de Kettle es fundamental entender la estructura, ya que tendremos que generar el contenido de los ficheros de forma dinámica, cambiando los datos de cada consulta. En el código 5.1 podemos ver el formato básico que tendrá el fichero.

```

1  <transformation>
2    <info>
3      [...]
4    </info>
5    <connection>
6      [...]
7    </connection>
8    [...]
9    <connection>
10     [...]
11   </connection>
12   <order>
13     <hop>

```

```

14     [...]
15     </hop>
16     [...]
17     <hop>
18     [...]
19     </hop>
20 </order>
21 <step>
22     [...]
23 </step>
24 [...]
25 <step>
26     [...]
27 </step>
28 </transformation>
29

```

Código 5.1: Estructura de archivo XML con extensión *.ktr* de transformación Kettle.

Como se puede observar en el código 5.1, el archivo XML está formado por una etiqueta raíz *transformation* que se divide en cuatro secciones principales:

- *info*: es la primera etiqueta con contenido que aparece. Dentro se definen aspectos generales de la transformación como el nombre.
- *connection (opcional)*: dentro de las etiquetas de estos elementos existe información de las conexiones con las BBDD.
- *order*: dentro de las etiquetas de este elemento se agrupan todos los enlaces o conectores (*hop*) entre los diferentes componentes (*step*) de la transformación.
- *step*: dentro de las etiquetas de este elemento se describen aspectos concretos de ese componente. Los componentes serían los descritos en la Subsubsección 6.5.6

5.3.3. Ejemplo mapeo de VWQL a KTR de Kettle

Los ficheros KTR son muchos y muy largos, por lo que vamos a poner un ejemplo de una consulta VWQL, el archivo resultante de forma visual en *Spoon* de PDI y el contenido de la transformación resultante estará en el Apéndice D. En la Sección 6 podremos ver el lenguaje VWQL completo.

La consulta de ejemplo (Código 5.2) que vamos a escoger es **checks**. Esta consulta extrae del chat de todos los juegos (al ser chat solo se aplica a HiddenRoom, Saturn y Coffeeshop), los mensajes que contengan uso de comprobaciones, de los jugadores *Alumno 1* y *Alumno 2*.

```

1 Evidence ejemplo:
2   get students "Alumno 1", "Alumno 2"
3   show checks
4   in all
5   .

```

Código 5.2: Ejemplo de consulta de EvalSim en lenguaje VWQL.

En las imágenes de las Figuras 5.4 y 5.5 se puede ver como quedaría la consulta realizada en EvalSim. Esta forma visual es la manera en la que se programan las transformaciones en Pentaho, pero EvalSim genera toda esa estructura de forma transparente al usuario, que solo ve el resultado de ejecutar esta transformación.

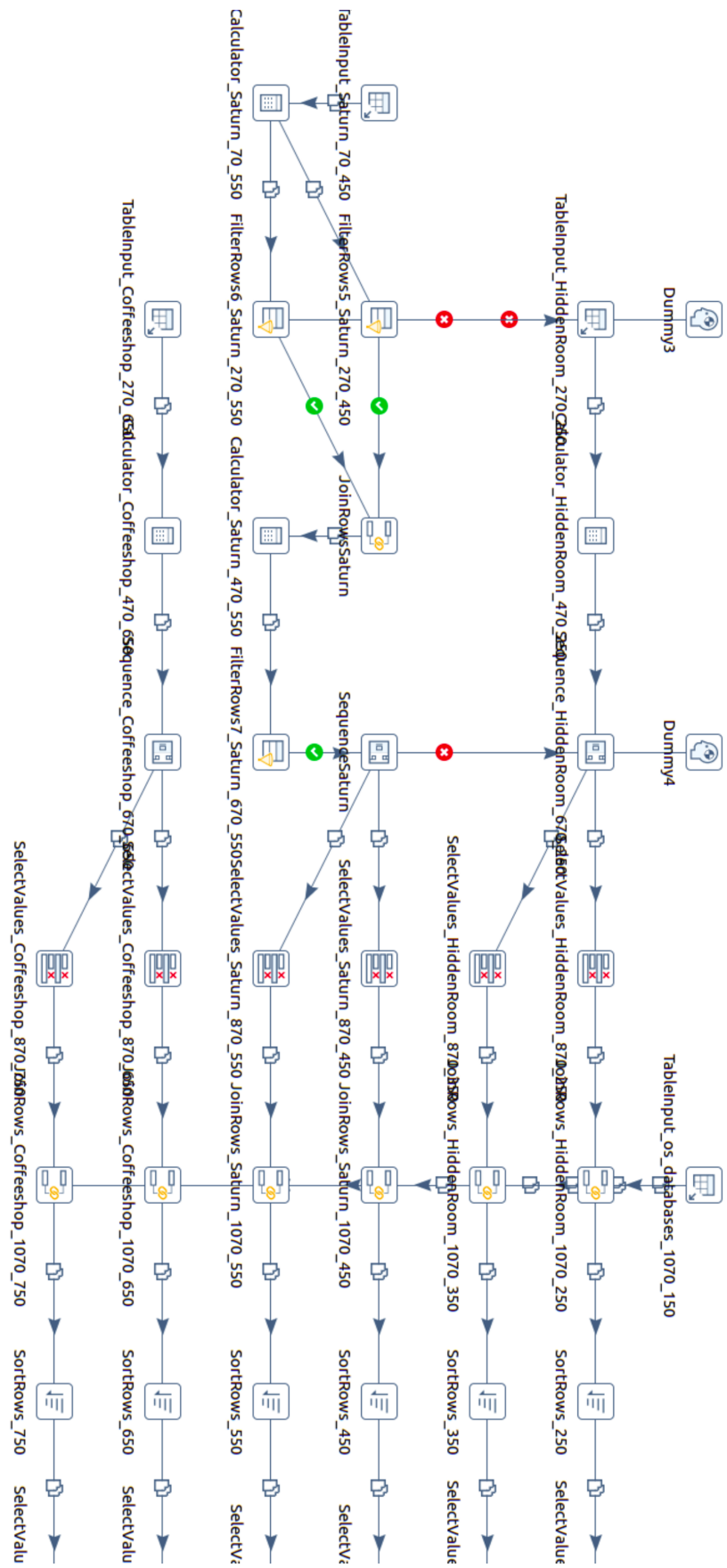


Figura 5.4: Imagen 1 de transformación de ejemplo con EvalSim.

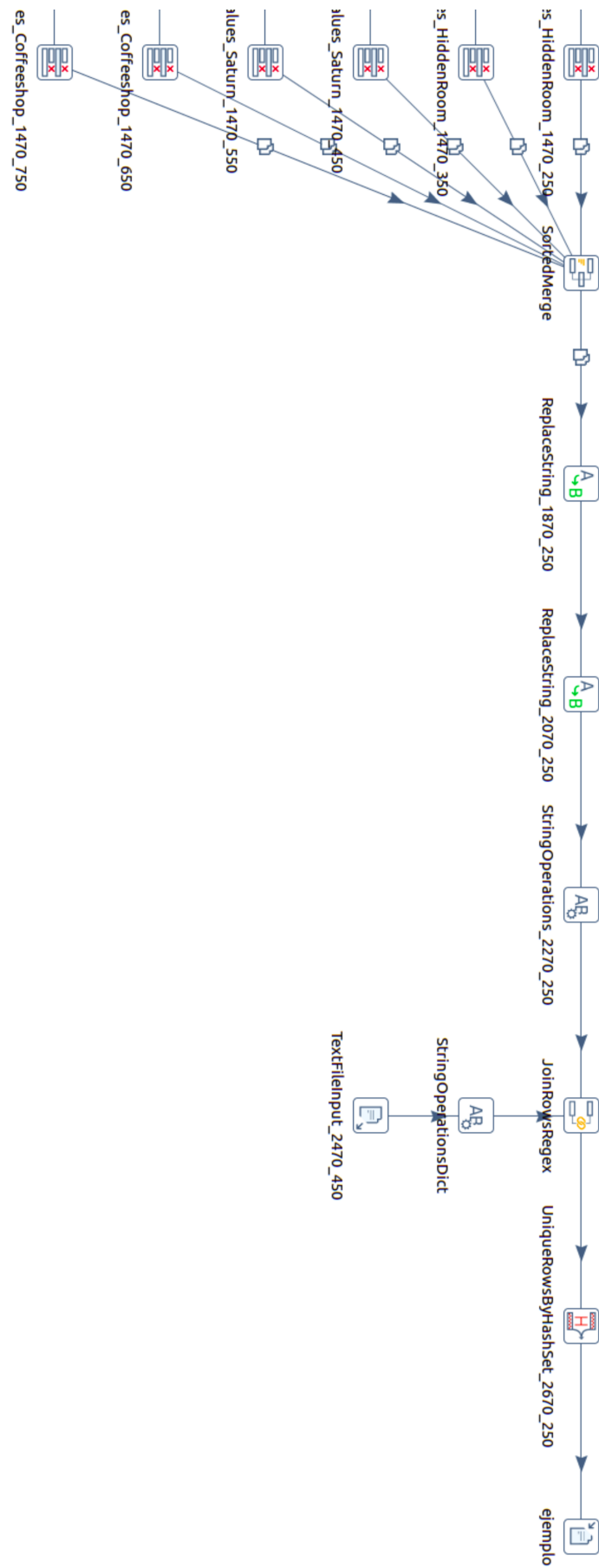


Figura 5.5: Imagen 2 de transformación de ejemplo con EvalSim.

Por último el resultado que muestra la ejecución de la consulta lo podemos ver en el Código 5.3

```
1 ID_GAME;NAME;REGION;DATE;MESSAGE
2 2;Alumno 2;HiddenRoom;2013/04/17 09:28:12;unter? oder auf!???
3 2;Alumno 2;HiddenRoom;2013/04/17 09:28:39;fertig?
4 2;Alumno 2;HiddenRoom;2013/04/17 09:30:39;fertig?
5 2;Alumno 2;HiddenRoom;2013/04/17 09:30:42;ist ok?
6 2;Alumno 2;HiddenRoom;2013/04/17 09:31:37;ok?
7 2;Alumno 2;HiddenRoom;2013/04/17 09:32:53;ok?
8 2;Alumno 2;HiddenRoom;2013/04/17 09:33:58;ok?
9 2;Alumno 2;HiddenRoom;2013/04/17 09:34:28;ok?
10 2;Alumno 2;HiddenRoom;2013/04/17 09:34:33;fertig?
11 2;Alumno 2;HiddenRoom;2013/04/17 09:35:10;Ok? weeee nummer eins
12 2;Alumno 2;HiddenRoom;2013/04/17 09:36:17;hinter?
13 2;Alumno 2;HiddenRoom;2013/04/17 09:36:45;fertig?
14 2;Alumno 2;HiddenRoom;2013/04/17 09:37:49;fertig?
15 2;Alumno 2;HiddenRoom;2013/04/17 09:37:52;ok?
16 2;Alumno 2;HiddenRoom;2013/04/17 09:39:47;fertig?
17 2;Alumno 2;HiddenRoom;2013/04/17 09:40:46;links?
18 2;Alumno 2;HiddenRoom;2013/04/17 09:40:49;oder rechts?
19 2;Alumno 2;HiddenRoom;2013/04/17 09:48:32;links oder rechts?
20 2;Alumno 2;HiddenRoom;2013/04/17 09:48:50;ok?
```

Código 5.3: Ejemplo de resultado de consulta en EvalSim.

5.4. Arquitectura de diseño

El patrón arquitectónico que se ha seguido para este proyecto ha sido *pipeline*. Este patrón arquitectónico consiste en una cadena de elementos, de tal forma que la salida de cada elemento es la entrada del siguiente elemento. Cada uno de éstos elementos va aplicando transformaciones al flujo de datos.

Los elementos transformadores del pipeline se van ejecutando en paralelo, por lo que entre los elementos se van almacenando temporalmente los datos.

Esta arquitectura presenta como ventajas:

- Permite comprender el comportamiento de entrada/salida del sistema como la composición del comportamiento de los elementos transformadores individuales.
- Facilita el crecimiento y mantenimiento del sistema.
- Soporta la ejecución de los flujos de datos de forma concurrente.
- Facilita la reutilización de transformaciones.
- Es fácil agregar o quitar transformaciones.

Sin embargo la arquitectura también presenta algunos inconvenientes:

- Puede complicarse al tener que mantener flujos separados pero relacionados.
- Puede ser necesario agregar elementos de transformación que normalicen los datos de entrada y salida, cuando los diferentes flujos tienen que converger a uno solo.
- Pérdida de rendimiento e incremento de complejidad de los elementos transformadores.

Al introducirse una consulta en EvalSim y dar a ejecutar el código, el sistema va pasando por las diferentes etapas del compilador hasta generarse el código de Kettle que posteriormente se ejecuta de forma automática, como se puede observar en la Figura .

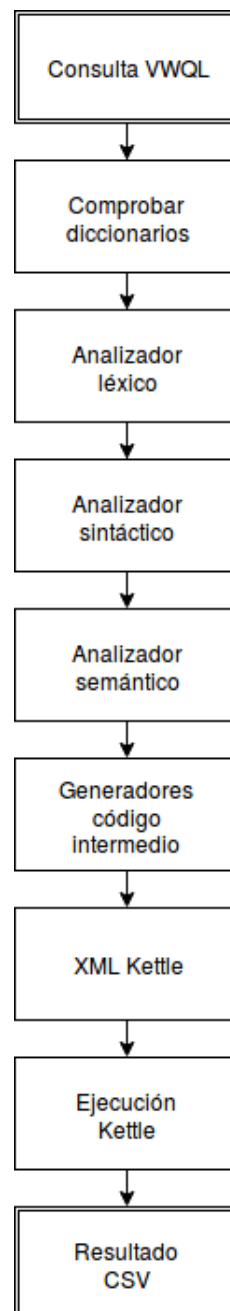


Figura 5.6: Diagrama de arquitectura de diseño.

5.5. Diseño interfaz de usuario

Al tratarse de un plugin, el aspecto visual es el de Eclipse. Aparte de esto, se ha generado un simple wizard para elegir el nombre del proyecto (Figura 5.7).

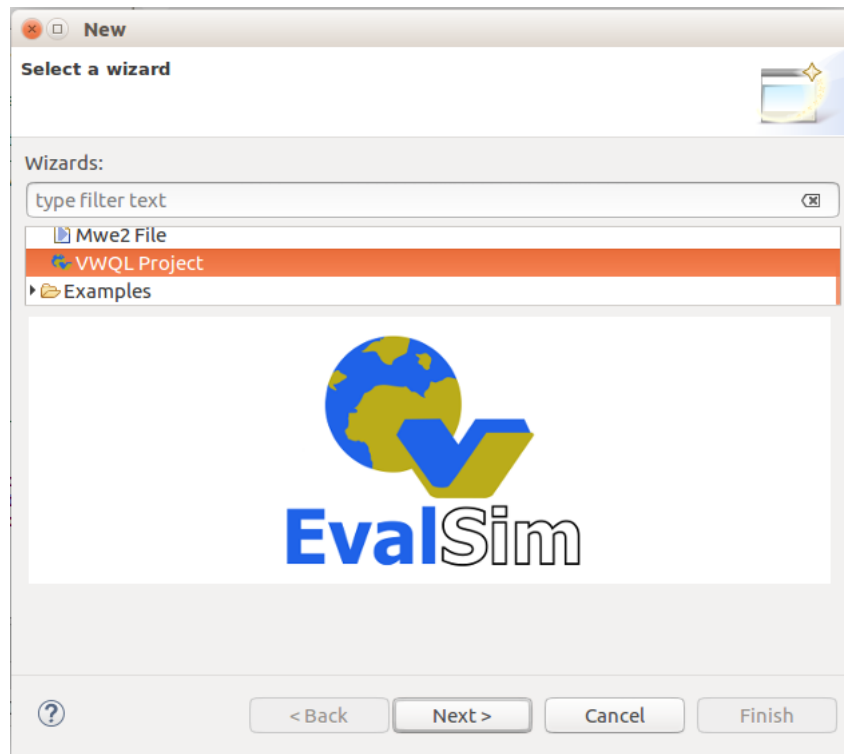


Figura 5.7: Captura del wizard de selección del proyecto.

Al terminar el wizard se abre el editor de EvalSim y se genera la estructura que se muestra en la Figura 5.8.

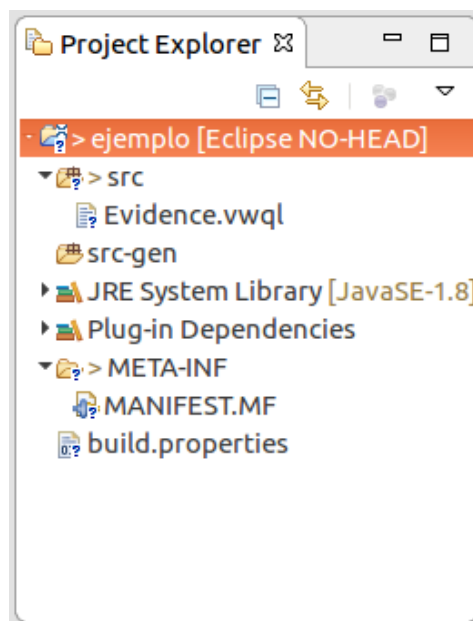


Figura 5.8: Captura de la estructura inicial del proyecto.

A continuación tenemos una imagen (Figura 5.9) de como quedaría el editor y la estructura al ejecutar una consulta en EvalSim. Para ejecutar la consulta, se abriría el menú de acción del botón secundario

del ratón sobre el editor de EvalSim, y se selecciona la opción *Generate Code*.

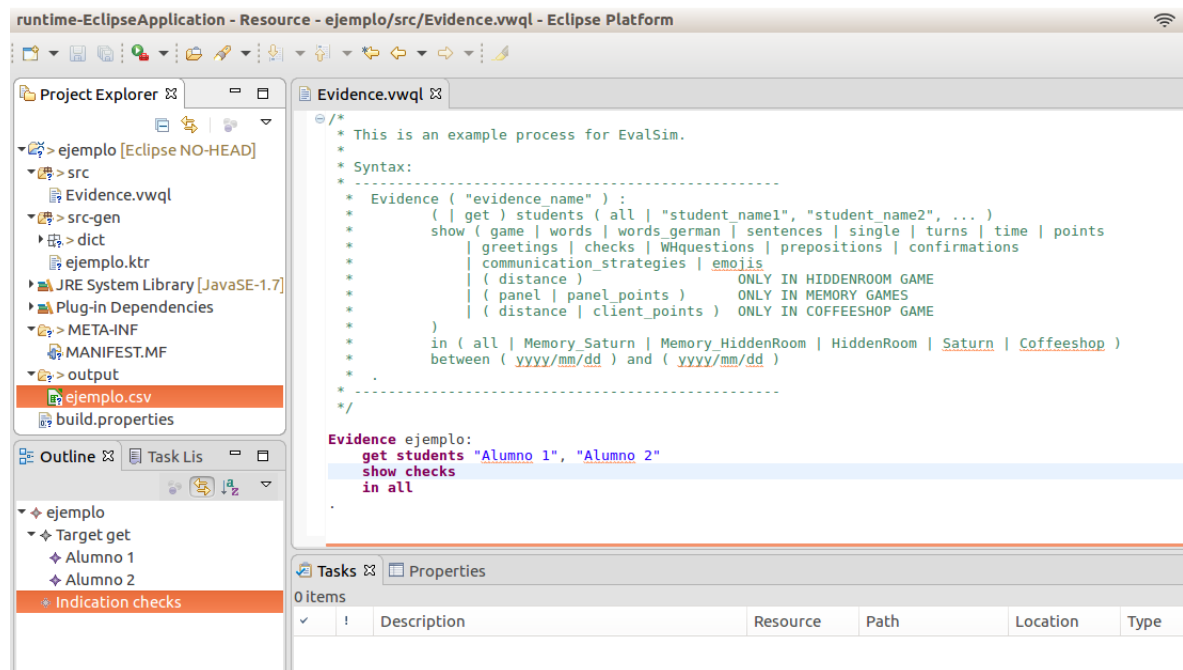


Figura 5.9: Captura de Evalsim.

En la Figura 5.10 se muestra en detalle como queda la estructura del proyecto tras ejecutar la consulta en EvalSim.

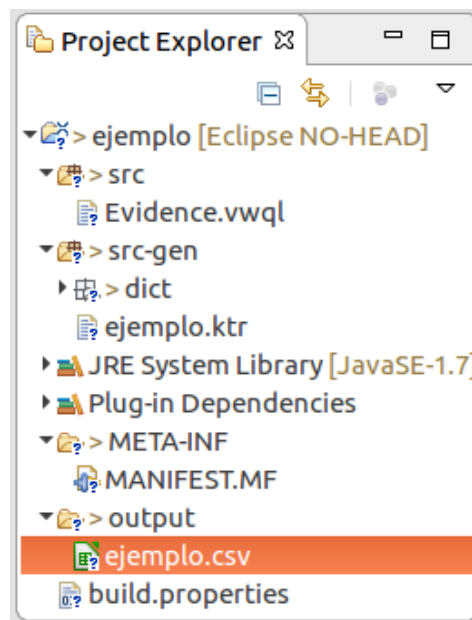


Figura 5.10: Captura de la estructura del proyecto después de ejecutar una consulta.

Capítulo 6

Implementación del Sistema

En este capítulo se procede a describir y explicar los detalles de implementación del sistema software. La implementación de nuestro DSL tiene tres pasos:

- *Metamodelo*: para implementar el modelo nos basaremos en lo mencionado en la Subsección 4.3.2. Se creará un metamodelo que recoja las características y restricciones del dominio del problema.
- *Gramática*: para la implementación de la gramática debemos crear un lenguaje a partir del metamodelo generado. Para ello definiremos su sintaxis y el modo en el que interactuarán los operadores y tokens.
- *Funcionalidad*: por último se deberá dotar de funcionalidad al lenguaje, es decir, que debe hacer EvalSim al escribir una consulta. En nuestro caso, que código se genera al introducir una consulta en nuestro lenguaje.

6.1. Metamodelo

Un DSL es un lenguaje de programación (de consulta en este caso) dedicado a resolver un problema particular, representando un problema específico y dando una solución al problema. Para representar el dominio del problema, haremos uso de un metamodelo, que dotaremos de conceptos del dominio a modelar, junto con las relaciones, reglas y restricciones aplicables a ese problema concreto.

Para crear el metamodelo usaremos el framework EMF [[Eclipse Modeling Framework, EMF](#)] y el proyecto Sirius [[Sirius](#)]. El proyecto Sirius nos permite crear un metamodelo con un editor gráfico. A medida que vamos creando el metamodelo de forma gráfica, Sirius va generando un archivo Ecore.

El archivo Ecore generado contiene el metamodelo, desarrollado de forma visual, en código XMI (XML Metadata Interchange). XMI es el nombre que recibe el estándar para el intercambio de metamodelos usando XML. Su principal objetivo es permitir un intercambio de metainformación entre herramientas de modelado basadas en UML y repositorios de metainformación basados en MOF en entornos distribuidos heterogéneos.

Para poder utilizar el metamodelo necesitamos las clases Java, y EMF nos proporciona una herramienta para generar las clases Java a partir de nuestro modelo Ecore. A través de la herramienta *Generator Model* y el archivo Ecore con el metamodelo, podemos generar automáticamente este código Java.

Para desarrollar el metamodelo crearemos un *Ecore Modeling Project* (EMP) como se muestra en la Figura 6.1.

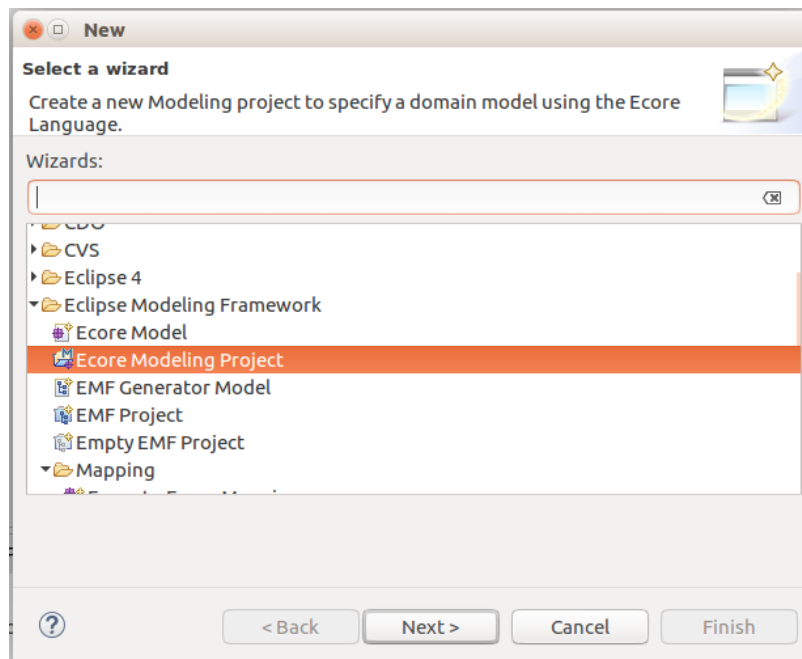


Figura 6.1: Paso 1: creación proyecto EMP.

Ahora debemos dar nombre al proyecto y elegir la ubicación (Figura 6.2). Cuando tengamos esto, le daremos a *Finalizar*.

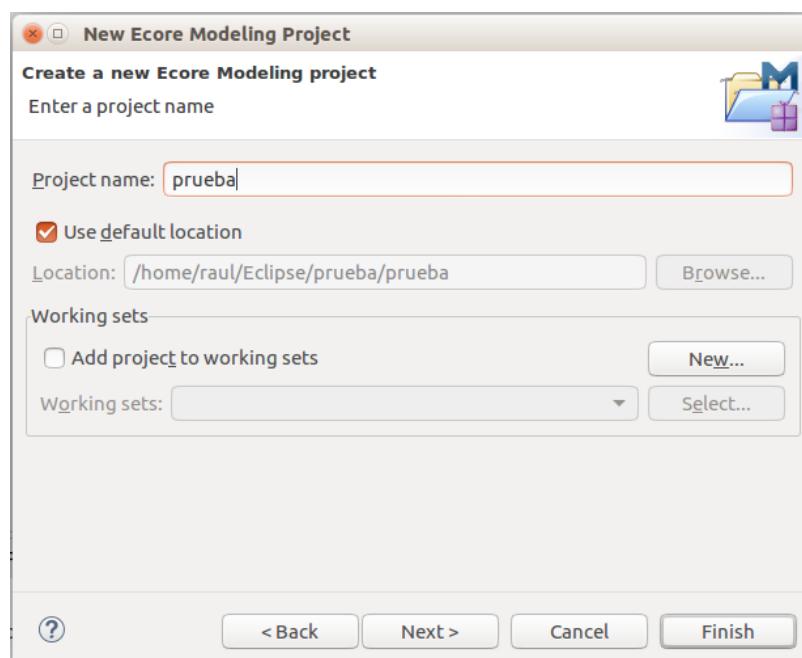


Figura 6.2: Paso 2: dar nombre y ubicación al proyecto.

En la Figura 6.3 podemos ver como quedaría en primera instancia el proyecto. Podemos observar la estructura en la parte izquierda de la figura y el editor de Sirius con los elementos que podemos seleccionar a la derecha.

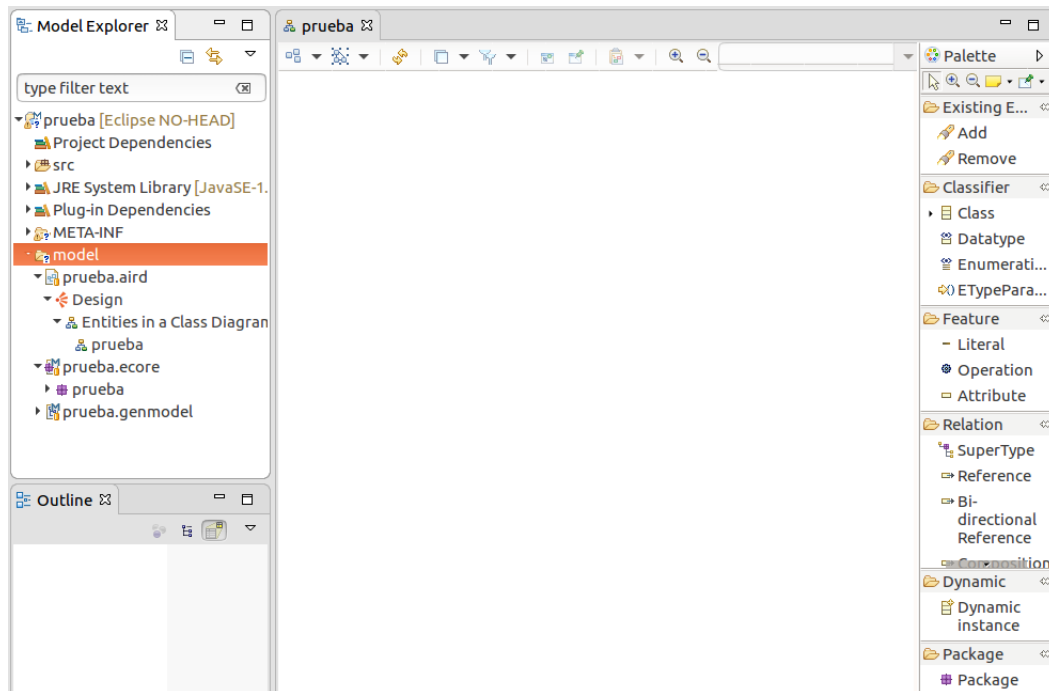


Figura 6.3: Paso 3: estructura del proyecto y editor visual de Sirius.

6.1.1. Metamodelo etapa 2

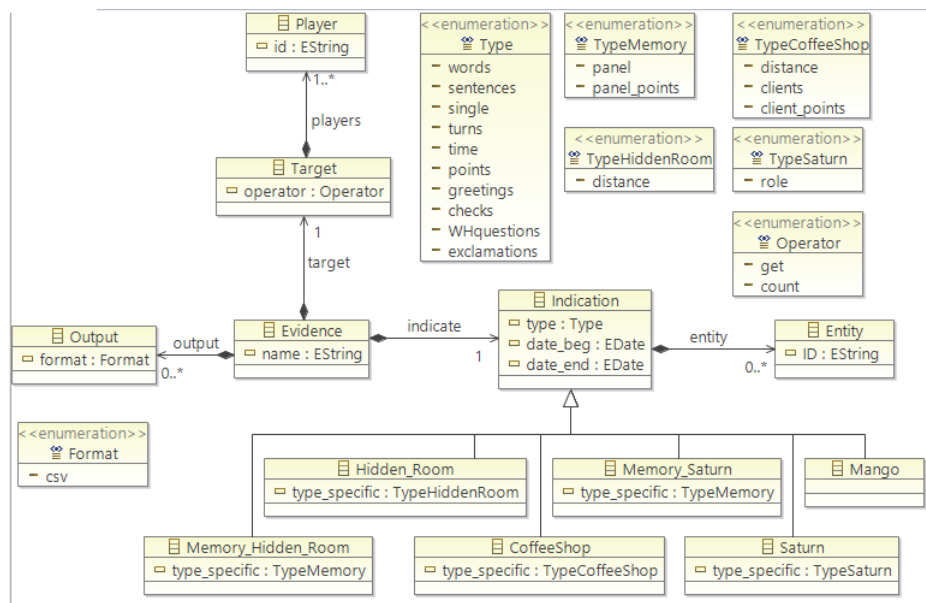


Figura 6.4: Diagrama del metamodelo en la etapa 2.

Como comentamos en el capítulo de planificación (Capítulo 2), el desarrollo del proyecto se ha dividido en etapas a través de las cuales el proyecto ha ido evolucionando. Un ejemplo de ello es el metamodelo, el cual podemos observar en la Figura 6.4.

El proyecto se empezó a crear en una versión más antigua de Eclipse, por tanto, con componentes desactualizados. También hay que tener en cuenta que el proyecto aun no estaba claramente definido, por lo que en esta segunda versión del metamodelo, aunque se aproxima a la versión actual, hay algunas cosas diferentes.

6.1.2. Metamodelo final

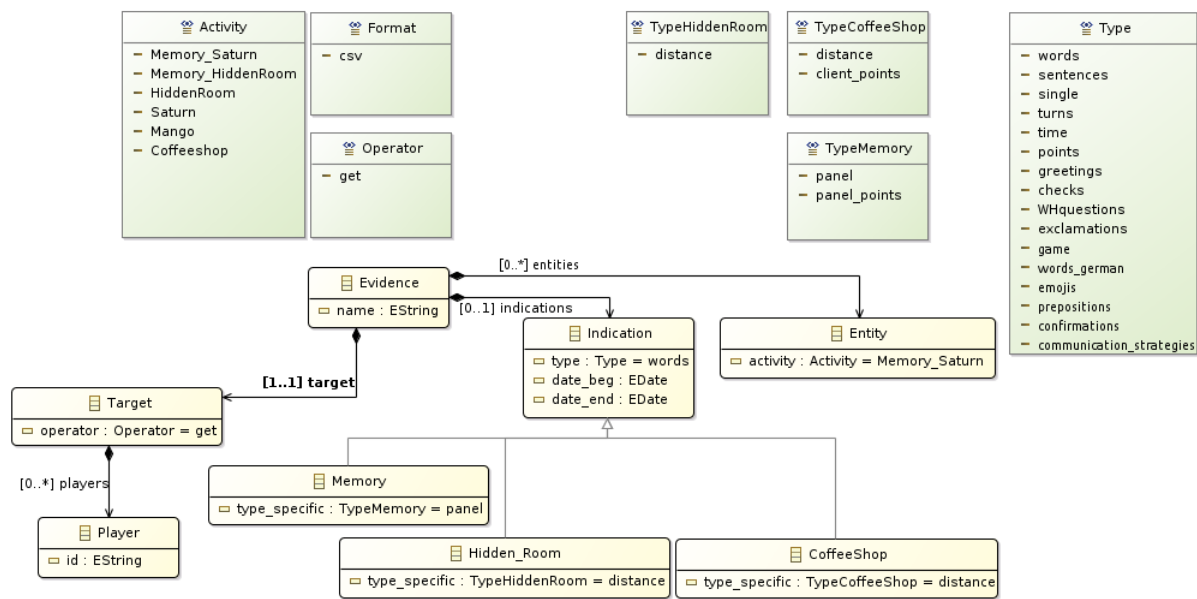


Figura 6.5: Diagrama de la versión definitiva del metamodelo.

El metamodelo de la Figura 6.5 es la versión final que se ha usado para desarrollar el DSL. Con este modelo se ha generado posteriormente la gramática que veremos en la siguiente sección.

El metamodelo ha sido desarrollado con la herramienta Sirius, así que, como hemos comentado anteriormente, se ha generado un archivo Ecore que contiene el metamodelo. La parte visual del metamodelo Ecore se puede ver en la Figura 6.6 y el código XMI se puede ver en el Apéndice E.

Una vez tenemos el metamodelo completo o parcialmente completo, podemos abrir el archivo *.genmodel* que se encuentra en la carpeta *model* de la Figura 6.3. Una vez abierto, le damos sobre el editor al botón *Generate All* y obtendremos el código Java del metamodelo. Con el código Java podremos continuar generando el DSL en los demás pasos.

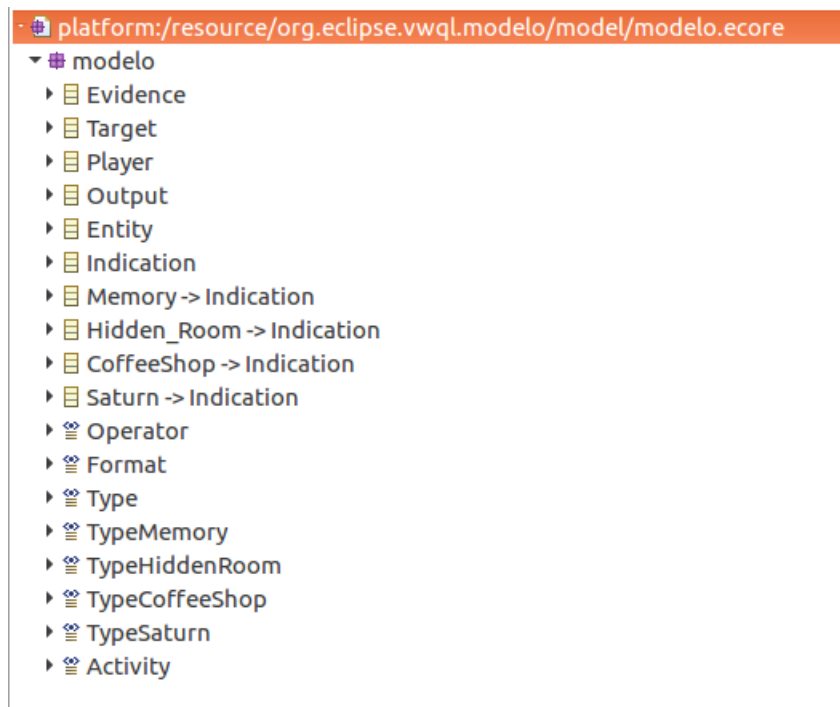


Figura 6.6: Contenido del fichero Ecore.

6.2. Gramática

Una vez tengamos el código Java del metamodelo, podemos comenzar el siguiente paso, crear la gramática. Para crear la gramática, el proyecto EMP nos proporciona la herramienta Xtext (Subsección 6.5.4).

Antes de desarrollar la gramática empezaremos por explicar como queremos que sea el lenguaje VWQL. En el Código 6.1 podemos ver el lenguaje completo, aunque necesita ser explicado.

Entre paréntesis podemos elegir entre las diferentes opciones que el lenguaje nos proporciona, o en algunos casos identificadores. Ahora pasamos a describirlo con más detalle, pudiendo dividir el lenguaje en 5 partes:

- En la primera línea del Código 6.1, podemos elegir un identificador que dará nombre a nuestra consulta y a los archivos que se generen. Después del identificador pondremos el token dos puntos, que indicará el inicio de la consulta
- En la segunda línea podemos elegir entre los diferentes operadores que queremos aplicar a la consulta. En este proyecto solo se ha dado funcionalidad al operador *get* que extrae la información del indicador elegido. La otra cosa que podemos elegir en esta línea es a que jugadores aplicar la consulta. Podemos elegir *all* para aplicarla a todos los jugadores de las BBDD; o elegir entre comillas el nombre/s del jugador/es a los que se va a aplicar.
- De las líneas 3 a la 5 podemos observar entre paréntesis muchos tokens. Debemos elegir un token entre todos ellos, que será el indicador que queremos extraer de los jugadores elegidos anteriormente. En lugar de elegir un token de aquí podemos elegir uno de las líneas 6, 7 u 8. No es posible seleccionar dos token a la vez.

- En las líneas 6, 7 y 8 podemos elegir un token, pero solo se aplicará a un juego específico, lo cual se indica a la derecha. No es posible seleccionar dos token a la vez.
- En la línea 10 podemos escoger de que juegos queremos extraer la información. Tenemos la opción de escoger varios de ellos o todos usando el token *all*.
- (OPCIONAL) En la penúltima línea podemos escoger el rango de fechas entre los que queremos obtener la consulta.
- En la última línea debemos colocar un punto, que será el token de cierre de la consulta.

```

1 Evidence ( "evidence_name" ) :
2   ( | get ) students ( all | "student_name1", "student_name2", ... )
3   show ( game | words | words_german | sentences | single | turns | time | points
4         | greetings | checks | WHquestions | prepositions | confirmations
5         | communication_strategies | emojis
6         | ( distance )           ONLY IN HIDDENROOM GAME
7         | ( panel | panel_points ) ONLY IN MEMORY GAMES
8         | ( distance | client_points ) ONLY IN COFFEESHOP GAME
9   )
10  in ( all | Memory_Saturn | Memory_HiddenRoom | HiddenRoom | Saturn | Coffeeshop )
11  between ( yyyy/mm/dd ) and ( yyyy/mm/dd )
12 .

```

Código 6.1: Lenguaje VWQL completo

Ahora que tenemos una idea más clara de como queremos que sea el lenguaje podemos pasar a la implementación de la gramática. Para ello, con la herramienta Xtext podemos generar una sintaxis del lenguaje por defecto, a partir de la cual podremos empezar a trabajar y modelar la sintaxis según nuestro planteamiento. En el Código 6.2, podemos ver el código en lenguaje Xtext que hemos implementado para nuestro DSL.

```

1 // automatically generated by Xtext
2 grammar org.eclipse.vwql.gramatica.VWQL with org.eclipse.xtext.common.Terminals
3
4 import "http://www.eclipse.org/emf/2002/Ecore" as ecore
5 import "platform:/resource/org.eclipse.vwql.modelo/model/modelo.ecore"
6
7 Evidence returns Evidence:
8   'Evidence'
9   name=EString
10  ':'
11   target=Target
12   ('show' indications=Indication)?
13   'in' ( 'all' | ( entities+=Entity ( "," entities+=Entity)* ) )
14   ('output' outputs+=Output ( "," outputs+=Output)* )?
15  ' ';
16
17 Target returns Target:
18   {Target}
19   operator=Operator
20   'students' (('all') | (players+=Player ( "," players+=Player)* ) )
21  ;
22
23 Output returns Output:
24   {Output}

```



```

25     format=Format
26     ;
27
28 Player returns Player:
29     id=EString
30     ;
31
32 Entity returns Entity:
33     {Entity}
34     activity=Activity
35     ;
36
37 Indication returns Indication:
38     Indication_Impl | Memory | Hidden_Room | CoffeeShop | Saturn;
39
40 Indication_Impl returns Indication:
41     {Indication}
42     (type=Type)
43     ('between' date_beg=EDate 'and' date_end=EDate)?
44     ;
45
46 Memory returns Memory:
47     {Memory}
48     (type_specific=TypeMemory)
49     ('between' date_beg=EDate 'and' date_end=EDate)?
50     ;
51
52 Hidden_Room returns Hidden_Room:
53     {Hidden_Room}
54     (type_specific=TypeHiddenRoom)
55     ('between' date_beg=EDate 'and' date_end=EDate)?
56     ;
57
58 CoffeeShop returns CoffeeShop:
59     {CoffeeShop}
60     (type_specific=TypeCoffeeShop)
61     ('between' date_beg=EDate 'and' date_end=EDate)?
62     ;
63
64 Saturn returns Saturn:
65     {Saturn}
66     (type_specific=TypeSaturn)
67     ('between' date_beg=EDate 'and' date_end=EDate)?
68     ;
69
70 Operator returns Operator:
71     'get' | 'count';
72
73 EString returns ecore::EString:
74     STRING | ID;
75
76 EDate returns ecore::EDate:
77     INT'-'INT'-'INT;
78
79 Format returns Format:
80     'csv';
81
82 Activity returns Activity:
83     'Memory_Saturn' | 'Memory_HiddenRoom' | 'HiddenRoom' | 'Saturn' | 'Mango' |
84     'Coffeeshop';

```

```

85 Type returns Type:
86     'words' | 'sentences' | 'single' | 'turns' | 'time' | 'points' | 'greetings' |
      'checks' | 'WHquestions' | 'exclamations' | 'game' | 'words_german' |
      'emojis' | 'prepositions' | 'confirmations' | 'communication_strategies' ;
87
88 TypeHiddenRoom returns TypeHiddenRoom:
89     'distance';
90
91 TypeMemory returns TypeMemory:
92     'panel' | 'panel_points';
93
94 TypeCoffeeShop returns TypeCoffeeShop:
95     'distance' | 'clients' | 'client_points';
96
97 TypeSaturn returns TypeSaturn:
98     'role';

```

Código 6.2: Gramática del lenguaje VWQL.

Una vez finalizada la gramática, debemos generar la infraestructura del lenguaje. Para ello seleccionamos con el botón secundario, sobre el fichero `.mwe2`, la opción *Run as ->Generate Xtext Artifacts*.

Al realizar esta acción se generarán los ficheros Java y una infraestructura de lenguaje ejecutable con un editor de Eclipse en el que utilizar nuestro lenguaje. Este editor contiene las siguientes funcionalidades: coloreado de sintaxis, autocompletado, validación rápida de la sintaxis.

6.3. Funcionalidad

Cuando tengamos toda la infraestructura del DSL y los ficheros Java, necesitamos dotar de funcionalidad al lenguaje. Recordemos que en primera instancia lo que debemos hacer es generar un fichero PDI que será ejecutado posteriormente. Para dar funcionalidad al lenguaje tendremos que especificar las reglas de generación de ficheros en función de la consulta que se realice.

Uno de los archivos que se crea al generar la infraestructura del lenguaje con Xtext, es un generador de código escrito en lenguaje Xtend (Subsección 6.5.5). Este fichero generador es el fichero principal donde especificaremos como se van a generar las consultas usando Xtend. El fichero se encuentra en el paquete `vwql.gramatica.generator` (Figura 6.7).

Las transformaciones de Pentaho se componen de varios elementos conectados a través de enlaces. En la subsección 5.3.1 del capítulo Diseño, hemos descrito los pasos que tienen las transformaciones que vamos a implementar. Cada uno de esos pasos se corresponden directamente con los elementos de Kettle que están descritos en la sección 6.5 de este capítulo.

Aunque el archivo *generator* es el principal, también nos ayudaremos de clases secundarias que nos facilitarán la implementación del código. Todos los archivos creados con esta finalidad se almacenarán en el paquete `vwql.gramatica.utils` (Figura 6.7). Los ficheros creados para generar las transformaciones son:

- *Classpath.xtend*: clase implementada para generar la estructura necesaria para crear una consulta.
- *DBconnections.xtend*: clase implementada para incluir en las transformaciones los elementos *connection* de PDI, que contiene información sobre las conexiones a las bases de datos.

- *Hop.xtend*: clase implementada para generar los conectores entre los elementos de las transformaciones.
- *Step.xtend*: clase implementada para generar los elementos PDI que aplicarán transformaciones al flujo de datos.
- *Xml.xtend*: clase implementada para generar la estructura XML del fichero de transformación PDI.

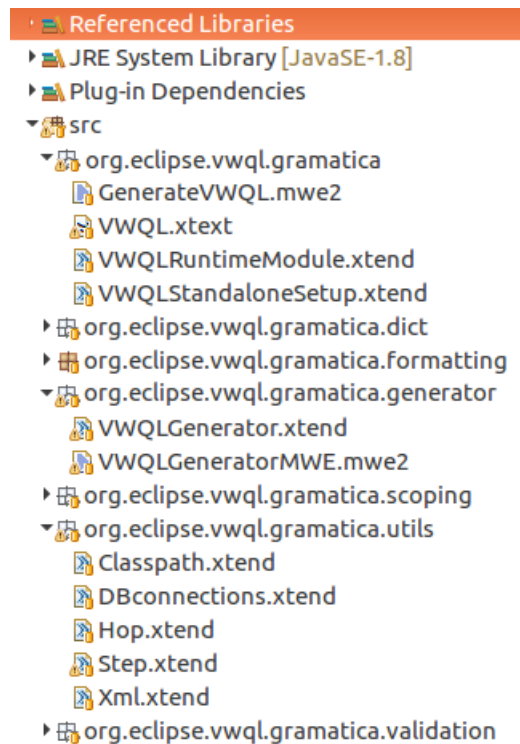


Figura 6.7: Estructura final del proyecto.

6.4. Control de versiones

Para controlar los cambios realizados en el sistema a lo largo del proyecto se ha usado *Git* ([Git]) a través del plugin *EGit* ([EGit]) de Eclipse. *Git* es un software de control de versiones diseñado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Algunas de las características más relevantes son:

- Rapidez en la gestión de ramas y mezclado de diferentes versiones. *Git* incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal.
- Gestión distribuida. *Git* le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.

- Realmacenamiento periódico en paquetes.

El IDE Eclipse tiene un excelente soporte para el sistema de control de versiones Git. Este soporte es provisto por el proyecto EGit a través de un conjunto de plugins.

Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Git. Bitbucket ofrece planes comerciales y gratuitos (hasta cinco usuarios en el caso de cuentas gratuitas).

- Disfruta de modelos de despliegue flexibles para equipos de cualquier tamaño y con todo tipo de necesidades. Alójalos en nuestra nube o gestiónalo en nuestros servidores.
- Repositorios privados u públicos ilimitados. Bitbucket Cloud es gratis para equipos de pequeños de 5 miembros.
- Los permisos de ramas proporcionan un control de acceso granular para su equipo, garantizando que las personas adecuadas pueden realizar los cambios correctos en su código.

El repositorio del proyecto se ha creado de forma privada, al que solo tiene acceso el personal relacionado con el mismo. El enlace al repositorio es [\[EvalSim\]](#)

6.5. Herramientas usadas

6.5.1. Eclipse Modeling Project (EMP)

El sistema EvalSim ha sido desarrollado a través del entorno tecnológico Eclipse Modeling Project (EMP) [\[Eclipse Modeling Project, EMP\]](#). EMP se centra en la evolución y promoción del desarrollo de tecnologías basadas en modelos de la comunidad Eclipse, aportando un conjunto integrado de herramientas, frameworks para modelaje y estándares de implementación.

6.5.2. Eclipse Modeling Framework (EMF)

En el núcleo de EMP, se encuentra Eclipse Modeling Framework (EMF) [\[Eclipse Modeling Framework, EMF\]](#). EMF es un framework de modelaje y facilidad de generación de código para construir herramientas y aplicaciones basadas en un modelo de datos estructurado. Desde una especificación del modelo descrita en XMI, EMF provee herramientas y apoyo para producir un conjunto de clases Java para el modelo, así como un conjunto de clases que permitan visualización y edición basándose en comandos del modelo y un editor básico. Los Modelos pueden ser especificados usando Anotación Java, documentos XML, o herramientas de modelado como Rational Rose, y después ser importados a EMF. Lo más importante de todo, EMF suministra las bases para la interoperabilidad con otras herramientas y aplicaciones basadas en EMF.

6.5.3. Ecore Tools

El componente Ecore Tools [\[EcoreTools\]](#) proporciona un entorno completo para crear, editar y mantener modelos Ecore. Este componente facilita la manipulación de modelos Ecore con un Editor Ecore gráfico y puentes con otras herramientas Ecore existentes.

Para generar nuestro metamodelo Ecore hemos usado Sirius. Sirius [\[Sirius\]](#) es un componente de Eclipse que permite crear fácilmente un workbench de modelado gráfico aprovechando las tecnologías de

Eclipse Modeling. Sirius ha sido creado para proporcionar un workbench genérico para la ingeniería de arquitectura basada en modelos que podría adaptarse fácilmente a las necesidades específicas.

Sirius genera un archivo XML con extensión *.ecore* a partir del metamodelo diseñado de forma gráfica.

En la Figura 6.8 podemos ver los componentes que conforman Ecore sin entrar en más detalles. Con estos componentes, en Sirius, se diseñará de forma gráfica nuestro metamodelo.

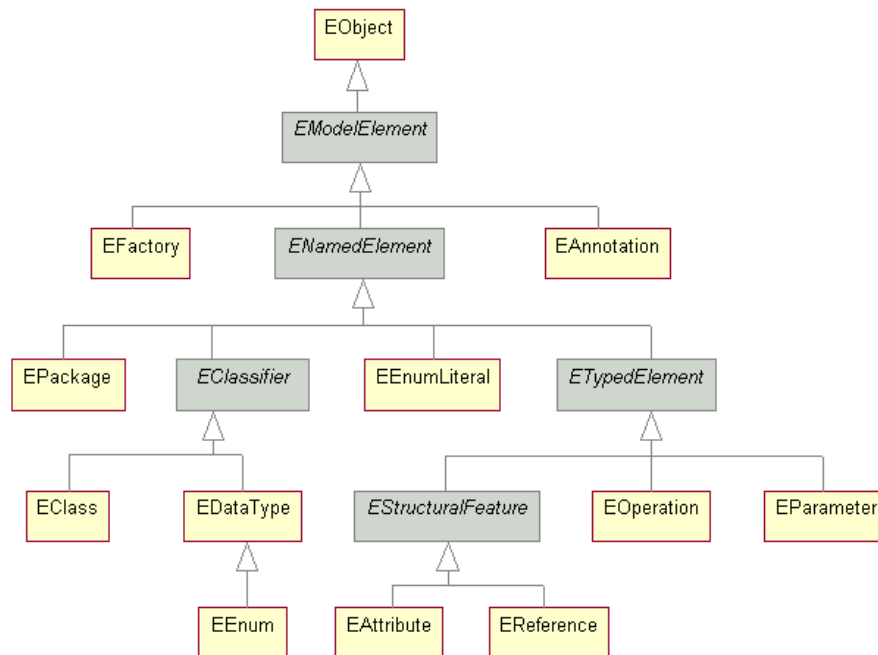


Figura 6.8: Jerarquía de componentes Ecore.

6.5.4. Xtext

Xtext [Bettini, 2016] es un framework de código abierto para el desarrollo de lenguajes de programación y lenguajes específicos de dominio (DSL). Xtext se está desarrollando como parte del proyecto EMF. Definiendo la gramática de nuestro lenguaje, Xtext es capaz de generar una infraestructura completa, que incluye un analizador sintáctico, un corrector ortográfico, un modelo de clases para el árbol de sintaxis abstracta y un IDE personalizable basado en Eclipse. [Xtext]

El framework ofrece una serie de características que facilitan la programación como:

- Coloreado de sintaxis.
- Autocompletado.
- Validación rápida de la sintaxis.
- Integración avanzada con Java.
- Integración con otras herramientas de Eclipse.

6.5.5. Xtend

Xtend [Bettini, 2016] es un lenguaje de programación de alto nivel de propósito general que es completamente interoperable con Java. Xtend tiene sintáctica y semánticamente sus raíces en el lenguaje de programación Java, pero se centra en una sintaxis más concisa y alguna funcionalidad adicional, como la inferencia de tipos, métodos de extensión, expresiones lambda y la sobrecarga de operadores. Todos los aspectos de la implementación del DSL en Xtext pueden ser implementados en Xtend en lugar de en Java, siendo más fácil de usar y permitiendo escribir un código más legible. Al ser interoperable con Java, se pueden usar todas las bibliotecas de Java. [Xtend]

6.5.6. Kettle (Pentaho Data Integration)

Pentaho Data Integration (PDI, también llamado Kettle) [Pentaho Data Integration, Kettle] es el componente de Pentaho responsable de los procesos de extracción, transformación y carga (ETL). Aunque las herramientas ETL se utilizan con mayor frecuencia en entornos de almacenes de datos, PDI también puede utilizarse para otros fines:

- Migración de datos entre aplicaciones o bases de datos.
- Exportación de datos de bases de datos a archivos de texto plano.
- Carga masiva de datos en bases de datos.
- Limpieza de datos.
- Integración de aplicaciones.

PDI es una herramienta de ETL de código abierto. PDI soporta una amplia gama de formatos de entrada y salida, incluyendo archivos de texto, hojas de datos y motores de base de datos comerciales y gratuitos. Además, las capacidades de transformación de PDI permiten manipular datos con muy pocas limitaciones.

Componentes Kettle

Cuando ejecutamos una consulta EvalSim, internamente el sistema está generando un archivo XML con extensión *.ktr*, que es lo que se denomina transformación en PDI. Por tanto, para poder generar este archivo primero debemos conocer cada uno de los componentes de PDI que necesitamos. En este apartado se mostrará cada componente Kettle usado.

- **Table input:** este paso es usado para leer información de una base de datos, usando una conexión y una consulta SQL. Una conexión es un atributo de configuración previamente establecido en el que se introducen los credenciales, el host y la base de datos a la que se quiere realizar la conexión.



- **Calculator:** este paso provee de funciones predefinidas que se aplican a los valores del campo de entrada.



- **Filter Rows:** este paso permite filtrar filas basándose en condiciones y comparaciones.



- **Java Filter:** este paso permite que el flujo de datos sea filtrado usando expresiones Java definidas por el usuario.



- **Group By:** este paso permite calcular valores definidos sobre grupos o campos.



- **Select Values:** este paso selecciona o modifica campos de una fila.



- **Sort Rows:** este paso ordena filas en base al campo que se le indique.



- **Text File Output:** este paso escribe filas en un fichero de texto, en nuestro caso en formato CSV separado por punto y coma.



- **Text File Input:** este paso permite leer datos desde diferentes tipos de archivos de texto. El formato más usado incluye campos separados por punto y coma generados por hojas de cálculo en formato de texto plano CSV.



- **Unique Rows:** este paso elimina filas duplicadas dejando ocurrencias únicas.



- **Sequence:** este paso añade una secuencia al flujo de datos.



- **Add Constant:** este paso añade una constante al flujo de datos.



- **Janino:** este paso permite introducir Clases Java Definidas por el Usuario para añadir un paso con funcionalidad personalizada.



- **Modified Java Script Value:** este paso permite realizar cálculos complejos usando.



- **Join Rows:** este paso permite producir combinaciones (producto cartesiano) de todas las filas en el flujo de datos de entrada.



- **Dummy:** este paso no hace nada. Su función principal es para testing.



- **Concat Fields:** este paso es usado para concatenar múltiples campos en uno solo.



- **Sorted Merge:** este paso mezcla filas procedentes desde múltiples pasos de entrada estando esta fila previamente ordenadas por los campos clave dados.



- **Split Fields to Rows:** este paso te permite dividir una fila que contiene un campo delimitado, en múltiples nuevas filas, una por cada valor obtenido al dividir la fila. La antigua fila no se envía a los pasos siguientes.



- **Replace in String:** este paso permite buscar y reemplazar. También soporta expresiones regulares.



- **String Operations:** este paso permite aplicar operaciones definidas a una cadena de texto.



- **Stream Lookup:** este paso permite buscar datos de un paso usando información procedente de otro paso. Los datos procedentes del paso fuente son leídos y almacenados primero en memoria y después son usados para buscar los datos desde el flujo principal.



- **Analytic Query:** este paso permite buscar datos en las filas anteriores y posteriores. Para ello las filas deben estar ordenadas.



6.5.7. MySQL

MySQL [MySQL] es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones. . .
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

6.5.8. L^AT_EX

L^AT_EX es un sistema de composición de textos, orientado a la creación de textos científicos y técnicos en especial.

L^AT_EX está formado por un gran conjunto de macros de T_EX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, T_EX. L^AT_EX se extendió rápidamente por todo el sector científico y técnico gracias a su facilidad de uso y toda la potencia de T_EX.

Su código abierto permitió que muchos usuarios realizasen nuevas utilidades que extendiesen sus capacidades con objetivos muy variados, apareciendo “dialectos” de L^AT_EX, muchas veces incompatibles entre sí. En 1993 se anunció una reestandarización completa de L^AT_EX para evitar discrepancias anteriores, creándose nuevas extensiones como la posibilidad de escribir transparencias por ejemplo.

La característica más relevante que se creó fue la arquitectura modular. Se estableció un núcleo central, el compilador, que mantiene las funcionalidades de la versión anterior pero permite incrementar su potencia y versatilidad por medio de diferentes paquetes, que cualquiera puede crear uno nuevo, que sólo

se cargan si son necesarios.

Esta memoria está desarrollada con \LaTeX . Además del diseño limpio y claro que proporciona al documento de manera automática, crea índices, referencias, bibliografía, ajusta figuras y listados, etc., y un sinfín de características que facilitan la labor generando además un documento impecable.

6.5.9. Draw.io

Draw.io es una aplicación online que permite elebaorar diagramas. Esta herramienta permite elaborar diagramas online, sin necesidad de instalar nada. Su interfaz es bastante sencilla y fácil de utilizar, además de ser bastante completa.

Sus opciones son todas las necesarias para elaborar completos diagramas. Dispone de una gran variedad de formas y diseños predeterminadas que luego podemos moldear a nuestro gusto. Permite agregar rápidamente imágenes externas utilizando el buscador de Google y tiene múltiples opciones de texto con formato que podemos configurar como mejor nos parezca.

Para utilizar esta aplicación no es necesario ningún tipo de registro, es totalmente gratuita y los trabajos realizados pueden ser guardados en formato .XML para su posterior modificación con la herramienta, podemos imprimirlos o también permite exportar los diagramas a formatos .PNG, .GIF, .JPG, .PDF y .SVG, o si preferimos podremos insertar el diagrama realizado en cualquier sitio web utilizando un código que nos genera la aplicación.

6.5.10. GIMP

GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la Licencia pública general de GNU.

GIMP lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, gif, png, pcx, tiff, y los de Photoshop, además de poseer su propio formato de almacenamiento de ficheros, xcf. También es capaz de importar ficheros en pdf y también imágenes vectoriales en formato svg creadas, por ejemplo, con Inkscape.

Capítulo 7

Pruebas del Sistema

En este capítulo se hablará de las pruebas que se han realizado en el sistema. La fase de pruebas es una parte importante en todo desarrollo software. El objetivo de las pruebas es la verificación de que el proyecto cumple con los requisitos especificados. Existen diferentes enfoques a la hora de realizar las pruebas de software.

- **Pruebas de integración:** Se realizan en el ámbito del desarrollo software una vez aprobadas las pruebas unitarias. Se refiere a la prueba de todos los elementos unitarios que componen un proceso realizado en conjunto. De esta forma conseguimos verificar que las partes de un sistema software funcionan de forma conjunta.
- **Pruebas funcionales:** Basadas en la ejecución, revisión y retroalimentación de las funcionalidades diseñadas para el software.
- **Pruebas de usabilidad:** Pruebas que se encargan de medir en que grado puede una persona usar el sistema. Consiste en seleccionar un grupo de usuarios y solicitarles que lleven a cabo las tareas para las cuales fue diseñada la aplicación, mientras que el equipo de desarrollo toma nota para evaluar la respuesta del usuario.
- **Pruebas de sistema y de aceptación:** Las pruebas de sistema aseguran que el sistema (tanto el software como el hardware) cumplan con todos los requisitos establecidos, mientras que las pruebas de aceptación validan que un sistema cumple con el funcionamiento esperado y permite al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

7.1. Pruebas unitarias

Permiten probar el correcto funcionamiento de un módulo de código. De esta forma conseguimos saber que cada uno de los módulos que integran el sistema software funcionan correctamente por separado.

7.1.1. Metamodelo

Cuando tenemos terminado el metamodelo podemos usar una herramienta de Eclipse a través de la cual podemos probar que el metamodelo es válido (Figura 7.1). Después con el asistente de Eclipse podremos automáticamente generar modelos conforme a dicho metamodelo.

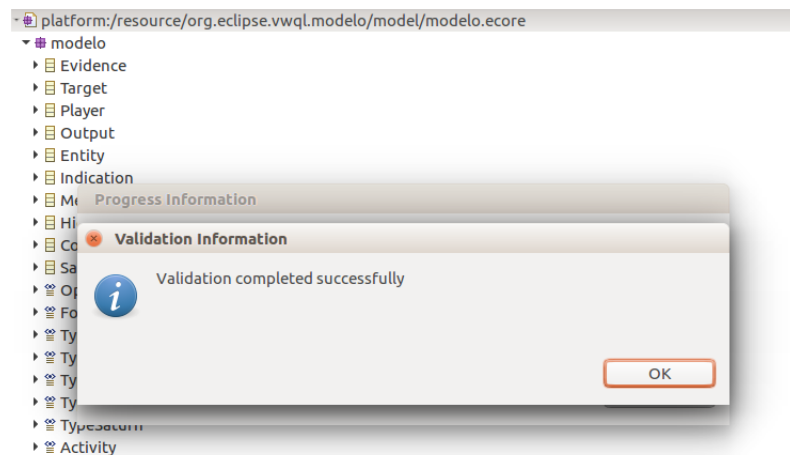


Figura 7.1: Prueba unitaria del metamodelo.

7.1.2. Editor

Cuando generamos la infraestructura del DSL de forma automática a través de la gramática, también se generó un editor de texto personalizado para nuestro DSL.

Este editor permite saber cuando hay errores en la sintaxis. En la Tabla 7.1 podemos ver que las consultas probadas conforme a la gramática devuelven el resultado esperado por el DSL.

Consulta	Verificación
<i>Evidence name: get students all show game in all</i>	✓
<i>Evidence name: get students all show points in MemorySaturn</i>	✓
<i>Evidence name: get students all show words_german in MemoryHiddenRoom</i>	✗
<i>Evidence name: get students all show words_german in HiddenRoom</i>	✓
<i>Evidence name: get students all show preposition in all</i>	✓
<i>Evidence name: get students all show panel in MemorySaturn</i>	✓
<i>Evidence name: get students all show panel in Saturn</i>	✗
<i>Evidence name: get students all show distance in HiddenRoom, Coffeeshop</i>	✓
<i>Evidence name: get students all show distance in MemorySaturn</i>	✗
<i>Evidence name: get students .Alumno 1"show checks in all</i>	✓

Tabla 7.1: Pruebas unitarias del editor.

7.1.3. Transformación

Cuando se realiza una consulta, el DSL genera un archivo con extensión *.ktr* que puede ser ejecutado con Kettle. Hemos probado todas las consultas anteriores en las que se esperaban resultados positivos (Tabla 7.2), y el programa PDI ha sido capaz de interpretarlos y ejecutarlos dando el resultado esperado.

Consulta	Verificación
Obtener partidas de todos los jugadores en todos los juegos	✓
Obtener puntos de todos los jugadores en MemorySaturn	✓
Obtener las palabras en alemán de todos los jugadores en HiddenRoom	✓
Obtener las preposiciones de todos los jugadores en todos los juegos	✓
Obtener los paneles jugados de todos los jugadores en MemorySaturn	✓
Obtener la distancia de todos los jugadores en HiddenRoom y en Coffeeshop	✓
Obtener los checks del jugador .Alumno 1. ^{en} todos los juegos	✓

Tabla 7.2: Pruebas unitarias de transformación.

7.1.4. Resultados

Como se ha comentado en la subsección anterior, los resultados de las consultas al ejecutar el archivo en Kettle han sido satisfactorios y se ha obtenido un fichero CSV con los datos esperados de cada una de las consultas probadas en la Tabla 7.2.

7.2. Pruebas de integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software. Los distintos módulos se fueron integrando hasta conseguir el funcionamiento del sistema completo. A continuación se muestra en la Tabla 7.3 la verificación de la interacción entre varios de sus componentes, pero será en el siguiente apartado, donde se vea el sistema funcionando al completo.

Módulos	Integración
Metamodelo - Editor (Xtext)	✓
Editor (Xtext) - Transformación (Xtend)	✓
Transformación (Xtend) - Resultados CSV	✓
Metamodelo - Editor - Transformación - Resultados CSV	✓

Tabla 7.3: Pruebas de integración de los módulos.

7.3. Pruebas de sistema y aceptación

En esta sección se realizan pruebas para asegurar que el sistema (tanto el software como el hardware) cumple con todos los requisitos establecidos: funcionales, de almacenamiento, reglas de negocio y no funcionales. Se suelen desarrollar en un entorno específico para pruebas, aunque nosotros al disponer de una copia de seguridad de las bases de datos, hemos realizado directamente las pruebas de sistema y aceptación a la vez.

7.3.1. Pruebas funcionales

Para la realización de las pruebas funcionales, seleccionaremos dos consultas que consideramos nos puede aportar información útil para investigación.

Uso de preposiciones

Con este indicador (Tabla 7.1) podemos conseguir todas las veces que un jugador ha usado correctamente preposiciones en alemán, obteniendo un listado de los mensajes en los que se ha detectado esta estructura

gramatical. Con este indicador podremos comprobar si los jugadores han adquirido correctamente los conocimientos necesarios para desenvolverse en el juego y en el idioma objetivo.

```
1 Evidence name:
2   get students "Alumno 1", "Alumno 2", "Alumno 3", "Alumno 4",
3   show preposition
4   in HiddenRoom
```

Código 7.1: Consulta VWQL.

En los juegos en los que se usa el chat, podemos comprobar el nivel de expresión que tienen los alumnos y la capacidad para relacionarse en el idioma objetivo. El juego HiddenRoom está diseñado específicamente para que los alumnos practiquen el uso de las preposiciones.

Como se puede apreciar en la Tabla 7.4 obtenemos información de los estudiantes seleccionados en la consulta. Podemos ver 5 campos en la tabla, que son identificador del juego (es generado), nombre del jugador, juego en el que se mandó el mensaje, fecha y hora del mensaje, mensaje. Podemos observar también como los mensajes tienen la estructura gramatical preposiciones.

Si observamos los jugadores, el Alumno 1 y el Alumno 3 claramente aparecen más en el indicador. Esto quiere decir que han usado bastante las preposiciones, y que los otros dos jugadores no la han usado demasiado. Esto se debe a que al ser un juego de rol cooperativo, un jugador se encargaba de decirle al otro que debía hacer y el otro ejecutaba las órdenes. Los jugadores Alumno 2 y Alumnos 4 también han usado el uso de las preposiciones pero en mucha menor medida.

Generalmente en este juego uno hablaba dando las órdenes y el otro las ejecutaba. Esto nos lleva a pensar que el segundo puede que haya usado más la estructura gramatical de comprobaciones por lo que haremos otra consulta de prueba para verlo en la siguiente subsubsección.

ID	NAME	REGION	DATE	MESSAGE
2	Alumno 1	HiddenRoom	2013/04/17 09:27:11	rechts auf dem regal
2	Alumno 1	HiddenRoom	2013/04/17 09:28:52	auf dem regal
2	Alumno 1	HiddenRoom	2013/04/17 09:29:41	der feuerzeuf auf dem regal
2	Alumno 1	HiddenRoom	2013/04/17 09:36:02	die flasche wein ist hinter dem salat und dem teller
2	Alumno 1	HiddenRoom	2013/04/17 09:39:20	un der feuerzeug ist auf dem kuchen
2	Alumno 1	HiddenRoom	2013/04/17 09:39:30	auf dem regal
2	Alumno 1	HiddenRoom	2013/04/17 09:40:29	auf dem regal
2	Alumno 1	HiddenRoom	2013/04/17 09:43:50	der rucksack ist unter dem bett
2	Alumno 1	HiddenRoom	2013/04/17 09:45:18	der Teddybär ist auf dem bett
2	Alumno 1	HiddenRoom	2013/04/17 09:49:37	der computer ist auf dem bett
2	Alumno 2	HiddenRoom	2013/04/17 09:49:45	auf dem bett
6	Alumno 3	HiddenRoom	2013/04/17 10:12:26	ja, und das glas Cola ist links neben der pizza
6	Alumno 3	HiddenRoom	2013/04/17 10:14:03	hinter dem teller pommes frites
6	Alumno 3	HiddenRoom	2013/04/17 10:16:07	ins links neben dem glas coka
6	Alumno 3	HiddenRoom	2013/04/17 10:19:50	ist links auf dem sessel
6	Alumno 3	HiddenRoom	2013/04/17 10:23:43	auf dem sofa
6	Alumno 3	HiddenRoom	2013/04/17 10:26:09	ist auf der waschmaschine
6	Alumno 3	HiddenRoom	2013/04/17 10:26:28	rechts neben der kaffeemaschine
6	Alumno 3	HiddenRoom	2013/04/17 10:28:01	auf dem sofa
6	Alumno 3	HiddenRoom	2013/04/17 10:30:13	ist auf dem sofa
6	Alumno 4	HiddenRoom	2013/04/17 10:30:22	wo auf dem sofa?
6	Alumno 3	HiddenRoom	2013/04/17 10:33:28	das vacuum ist links neben dem sofa
6	Alumno 3	HiddenRoom	2013/04/17 10:34:58	die lampe ist links neben dem bett

Tabla 7.4: Pruebas de funcionales, resultado de consulta *prepositions*.

Uso de comprobaciones

Con este indicador (Código 7.2) podemos extraer todos los mensajes de los alumnos en los que haya una de las estructuras de confirmación. Entendemos por esto que quiere decir que si un jugador dicta una orden, el otro jugador responderá con un mensaje de confirmación de que lo ha entendido o que ha realizado la acción. Con este indicador podremos comprobar si los jugadores han adquirido correctamente los conocimientos necesarios para desenvolverse en el juego y en el idioma objetivo. También podemos comprobar si se entienden entre ellos y por ende, si existe comunicación.

```

1 Evidence name:
2   get students "Alumno 1", "Alumno 2", "Alumno 3", "Alumno 4",
3   show checks
4   in HiddenRoom

```

Código 7.2: Consulta VWQL.

Como se ha mencionado antes, en los juegos en los que se usa el chat, podemos comprobar el nivel de expresión que tienen los alumnos y la capacidad para relacionarse en el idioma objetivo. Como el juego está pensado de forma colaborativa en la que uno ordena hacer algo y el otro ejecuta la orden, en la primera consulta hemos comprobado las preposiciones que indican que tiene que hacer el segundo jugador, y en esta segunda consulta las comprobaciones de que este segundo jugador lo ha hecho bien.

Como se puede apreciar en la Tabla 7.5 obtenemos información de los estudiantes seleccionados en la

consulta. Podemos ver 5 campos en la tabla, que son identificador del juego (es generado), nombre del jugador, juego en el que se mandó el mensaje, fecha y hora del mensaje, mensaje. Podemos observar también como los mensajes tienen la estructura gramatical checks.

Si observamos los jugadores en la tabla, el Alumno 2 y el Alumno 4 claramente aparecen más en el indicador. Esto quiere decir que han usado bastante las comprobaciones, y que los otros dos jugadores no han usado esta estructura.

ID	NAME	REGION	DATE	MESSAGE
2	Alumno 2	HiddenRoom	2013/04/17 09:28:12	unter? oder auf!???
2	Alumno 2	HiddenRoom	2013/04/17 09:28:39	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:30:39	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:30:42	ist ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:31:37	ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:32:53	ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:33:58	ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:34:28	ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:34:33	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:35:10	ok? weeee nummer eins
2	Alumno 2	HiddenRoom	2013/04/17 09:36:17	hinter?
2	Alumno 2	HiddenRoom	2013/04/17 09:36:45	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:37:49	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:37:52	ok?
2	Alumno 2	HiddenRoom	2013/04/17 09:39:47	fertig?
2	Alumno 2	HiddenRoom	2013/04/17 09:40:46	links?
2	Alumno 2	HiddenRoom	2013/04/17 09:40:49	oder rechts?
2	Alumno 2	HiddenRoom	2013/04/17 09:48:32	links oder rechts?
2	Alumno 2	HiddenRoom	2013/04/17 09:48:50	ok?
3	Alumno 4	HiddenRoom	2013/04/03 10:43:38	is das korrekt?
6	Alumno 4	HiddenRoom	2013/04/17 10:15:44	ist das ok?
6	Alumno 4	HiddenRoom	2013/04/17 10:27:08	ist das korrekt?
6	Alumno 4	HiddenRoom	2013/04/17 10:30:27	links?
6	Alumno 4	HiddenRoom	2013/04/17 10:34:04	ist das jetzt okay?

Tabla 7.5: Pruebas de funcionales, resultado de consulta *checks*.

Capítulo 8

Conclusiones

En este último capítulo se detallan las conclusiones extraídas y la experiencia personal tras el desarrollo del presente proyecto. También se detallarán las posibles mejoras sobre el software desarrollado.

8.1. Objetivos

En esta sección veremos si se ha conseguido cada uno de los objetivos que se presentaron en el capítulo de Introducción.

Objetivo principal

- **Creación de DSL.** Se ha conseguido crear un *DSL* que extraiga información útil e importante para el investigador sobre la interacción y comportamiento de los alumnos en los mundos virtuales. Se han desarrollado bastantes indicadores que extraen información útil de los mundos virtuales y que ayudarán al proceso de investigación.
- **Facilidad de uso.** Se ha conseguido que el software sea muy fácil de usar, de forma que cualquier usuario pueda utilizarlo con sencillos pasos. Con solo realizar una consulta y darle a la opción *Generate Code* se consigue que se cree un archivo Kettle de transformación y se ejecute, todo ello de forma transparente al usuario, devolviendo un CSV con el listado del indicador consultado.

Objetivos secundarios

- **Sintaxis sencilla.** Se ha desarrollado un *DSL* con una sintaxis tan sencilla e intuitiva que con solo una pequeña guía, cualquier investigador pueda usarla.
- **Metamodelo genérico.** Se ha desarrollado un metamodelo lo más genérico posible para el *DSL*, de forma que sea válido para posibles expansiones del mundo virtual y admita la integración de nuevos indicadores. Se pueden agregar de forma sencilla nuevos indicadores de forma general a todos los juegos y de forma específica para un juego concreto.
- **Investigar indicadores.** Se ha conseguido implementar bastantes consultas de indicadores, de los juegos y del idioma en cuestión (alemán), para obtener que información relevante que pueda ser objeto de estudio en investigación. Todo ello se ha conseguido consultando a los profesores expertos en las materias, para conocer su opinión y saber que indicadores consideran aptos para ser estudiados.
- **Investigar Kettle de Pentaho.** Se ha estudiado el funcionamiento de *Kettle* para generar las consultas de los indicadores, y también la estructura del fichero para que el *DSL* pueda generar un

fichero válido que pueda ser ejecutado por el propio *Kettle*. Todo ello gracias a realizar muchos ejemplo en PDI, estudiar la estructura de los archivos *.ktr* y probando algoritmos de construcción de los ficheros de forma automática.

- **Integración del Software.** Se ha conseguido parcialmente integrar el software como un todo: Eclipse, EvalSim y Kettle; para facilitar al usuario la implantación del software. Se ha desarrollado un plugin que se instala en el entorno Eclipse, y que se integra en este IDE. El objetivo era un poco más ambicioso ya que se pretendía conseguir una aplicación RCP, que contuviese solo el plugin.

8.2. Dificultades

Durante la realización del proyecto se han tenido bastantes dificultades en las que había ocasiones que se podían evitar y otras que causaron bloqueo total del desarrollo. A continuación los explicaremos:

8.2.1. Desarrollo de aplicación

Al principio del proyecto contábamos con la ayuda del director del proyecto, y una documentación que era de gran utilidad y que iba a facilitar bastante el desarrollo del DSL. EL problema es que a medida que avanzábamos nos dimos cuenta de que algunos de los componentes Eclipse de esa versión que se usaban estaban desactualizados, lanzaban errores, no tenían soporte. . .

Se optó entonces por cambiar a una versión mas actual de Eclipse que tuviese los componentes actualizados. El problema era que también había que empezar a actualizar algunas partes del desarrollo.

Uno de estos cambios fue pasar de em Ecore Diagram a *Sirius*. Al tener que cambiar el editor del metamodelo, tuvimos que empezar a estudiar la tecnología y a copiar uno a uno los componentes que se habían desarrollado. Las diferencias entre *Ecore Diagram* y *Sirius* son como generar el proyecto:

- En *Ecore Diagram* teníamos que generar un proyecto EMF vacío, crear un archivo Ecore dentro del proyecto y un *Ecore Diagram* asociado a al archivo Ecore.
- En *Sirius* solo hay que crear un proyecto EMP (*Ecore Modeling Project*), *seleccionando el viewpoint de Design y activar la perspectiva Modeling. Si no se activa la perspectiva Modeling, no carga el archivo contenido en Design y no se puede abrir el editor visual de Sirius, con lo cual no se puede crear el metamodelo.*

8.2.2. Estudio e integración de Kettle

Otro retraso en el proyecto lo causó el estudio de la tecnología Kettle y su integración en Eclipse. A continuación detallamos un poco más los puntos que ocasionaron los retrasos:

- La labor de investigación de la tecnología presentó la dificultad de que no se había trabajado nunca con la arquitectura pipeline, y que Kettle trae muchos elementos para transformación del flujo de datos.
 - A medida que se iban definiendo los indicadores, teníamos que buscar los componentes adecuados que nos permitían tratar los datos. En ocasiones hay componentes similares, que pueden servir o no para realizar la transformación; o componentes que para nuestro caso particular estaban optimizados.
 - También hay que recordar que al tratarse de arquitectura pipeline, los flujos de datos paralelos tienen que normalizarse si queremos que converjan a un solo flujos de datos.

- En el mapeo e integración de los componente Kettle en el DSL no hubo grandes dificultades, pero si requiere una inversión de tiempo considerable. Al tratarse de transformaciones grandes, con procesamiento de datos y normalización de flujos de datos, se usaban muchos componentes con diferentes parámetros a tener en cuenta. Esto hace que la implementación conlleve bastante tiempo.
- El mayor retraso de todo el proyecto fue la integración de las librerías de PDI en el DSL para poder ejecutar la transformación de forma automática. Este paso se empezó a investigar mucho antes de la fecha que pone en el diagrama Gantt, debido a que teníamos conocimientos de que podían surgir problemas. Se empezó a estudiar en febrero y hasta finales de julio no se consiguió. Desde febrero hasta junio no hubo problemas porque aunque se seguía investigando, se podía continuar con el desarrollo del proyecto haciendo este paso de forma manual. Entre Junio y Julio se ocasionó un periodo de bloqueo del proyecto debido a este problema.

8.2.3. Estudio e integración de diccionarios

Al igual que en el caso de antes, integrar los diccionarios dentro del plugin ocasionó algunos retrasos, aunque bastante más leves que los sufridos por la integración de Kettle.

También se comentó antes que necesitábamos buscar un componente de Kettle que aplicase la transformación adecuada al flujo de datos de forma óptima. Un ejemplo de este caso es comparar cadenas de textos que provienen de dos flujos diferentes. PDI nos trae varios componentes con los que realizar esta tarea, de manera que se probaron diferentes componentes, y se modificaron las consultas e incluso las transformaciones integradas en el DSL. Finalmente se encontró el mejor componente para esta tarea.

8.2.4. Facilidad de uso

La facilidad de uso es uno de los objetivos principales del proyecto, y ha ocasionado que se tenga que modificar varias veces el metamodelo, con el consiguiente cambio de la gramática y de las funcionalidades desarrolladas para la gramática.

8.3. Experiencia personal

Personalmente, la elaboración tanto del proyecto como de la documentación ha supuesto un gran reto. Es una nueva experiencia que he tenido que afrontar sin haber hecho ningún sistema parecido hasta el momento. Los trabajos realizados en años anteriores, de las diferentes asignaturas de la carrera, han sido de menor duración y dificultad. Enfrentarte a un un proyecto de estas características supone un cambio bastante importante en la perspectiva de la carrera, porque supone vivir a pequeña escala la experiencia del mundo laboral. Hasta ahora, lo visto en las asignaturas estaba principalmente enfocado a adquirir conocimientos teóricos y prácticos, pero siempre los profesores sabían que tenías que hacer y como tenías que hacerlo. Este proyecto ha supuesto tener que construir desde una idea una herramienta que no se sabía en un principio como iba a funcionar, pero que poco a poco fue cogiendo forma hasta obtener el producto final.

También ha sido difícil aprender como funcionan las tecnologías, herramientas y los lenguajes de programación usados en el proyecto, que no conocía hasta el momento. Aprender a manejarlos y sacarles provecho ha supuesto perder el miedo a trabajar con herramientas y tecnologías desconocidas.

Aunque el proyecto me ha acercado al mundo laboral, siempre he tenido ayuda por parte de los investigadores Anke Berns, Manuel Palomo y Juan Manuel Doderó, junto con el director del proyecto Antonio Balderas. Les estoy muy agradecidos porque me han prestado toda la ayuda posible, me han dado la oportunidad de trabajar en un proyecto de investigación como este durante varios años y han hecho que la elaboración del proyecto resulte más fácil.

8.4. Trabajo futuro

La investigación es considerada una actividad orientada a la obtención de nuevos conocimientos y su aplicación para la solución a problemas o interrogantes. Por su parte la investigación docente es considerada como una modalidad de desarrollo profesional continuo que permite a los profesores identificar soluciones a problemas educativos y cambiar sus esquemas y percepciones sobre su labor educativa.

Después de desarrollar los mundos virtuales con los diferentes juegos, hay que evaluar si este método ayuda a los alumnos a realizar un aprendizaje autónomo.

El software desarrollado en este proyecto nos facilita la labor de obtención de resultados para comprobar si hemos conseguido una solución correcta al aprendizaje autónomo de ciertas competencias a través de cada uno de los juegos desarrollados en los mundos virtuales.

Los mundos virtuales dan muchas posibilidades para crear juegos y actividades diferentes que ayuden al aprendizaje autónomo de nuevas competencias. Posteriormente debe evaluarse si con un juego concreto se consigue el objetivo deseado.

EvalSim se ha desarrollado para adaptarse a los cambios y a las nuevas actividades que se desarrollen dentro de los mundos virtuales, como:

- Incluir nuevos indicadores en los que no se había pensado.
- Modificar o incluso eliminar indicadores ya desarrollados en el DSL.
- Incluir nuevos juegos con sus indicadores específicos.
- Incluir nuevos operadores que nos muestren los datos de otra forma. Por ejemplo gráficas o un operador que cuente los datos.
- Incluir operadores que nos comparen dos o mas indicadores dentro del propio DSL.

Apéndices

Apéndice A

Manual de usuario

Instrucciones de uso del sistema EvalSim.

A.1. Introducción

EvalSim es un DSL creado para ayudar a la investigación docente en la obtención de indicadores para evaluar las competencias y conocimientos adquiridos por los alumnos en el aprendizaje de idiomas, en concreto Alemán nivel A1. Estos indicadores se extraerán de la interacción de los alumnos con el entorno virtual y con otros alumnos, a través de sus avatares en mundos virtuales.

El sistema permite realizar consultas en el lenguaje VWQL, a las bases de datos de la plataforma de mundos virtuales desarrollada por la Universidad de Cádiz. Los mundos virtuales han sido desarrollados con el software OpenSim.

A través de la plataforma de mundos virtuales se generan gran cantidad de datos sobre la interacción que se produce entre jugadores, y entre jugadores y el entorno virtual. Analizar esta gran cantidad de datos es un proceso largo, que conlleva que un administrador de la plataforma extraiga los datos, los procese y se los envíe a los investigadores.

El sistema EvalSim permite a los investigadores realizar procesos ETL (Extracción, Transformación y Carga) complejos y tediosos, a través de una simple consulta en lenguaje VWQL.

Es una herramienta que ahorra tiempo y trabajo a la investigación docente. Directamente el investigador puede acceder a los datos ya procesados, y solo necesita visualizarlos y analizarlos.

A.2. Características

El sistema EvalSim permite las siguientes funcionalidades:

- Consultas a los juegos o actividades existentes dentro de los mundos virtuales:
 - MemoryHiddenRoom
 - MemorySaturn
 - HiddenRoom
 - Coffeeshop
 - Saturn

■ Indicadores que se pueden extraer de los juegos o actividades:

- game
- words
- words_german
- sentences
- single
- turns
- time
- points
- greetings
- checks
- WHquestions
- prepositions
- confirmations
- communication_strategies
- emojis
- distance (Solo en juegos HiddenRoom y Coffeshop)
- panel (Solo en juegos Memory)
- panel_points (Solo en juegos Memory)
- client_points (Solo en juego Coffeshop)

A.3. Requisitos previos

Requisitos software:

- Eclipse
- Herramienta EMF
- Pentaho (solo si se desea usar consultas fuera de EvalSim.)

A.4. Utilización

Para usar el sistema debemos abrir Eclipse y crear un proyecto EvalSim. Para crearlo vamos a *Nuevo* -> *Proyecto* y en el listado aparecerá *VWQL Project* con el logotipo de EvalSim (Figura A.1).

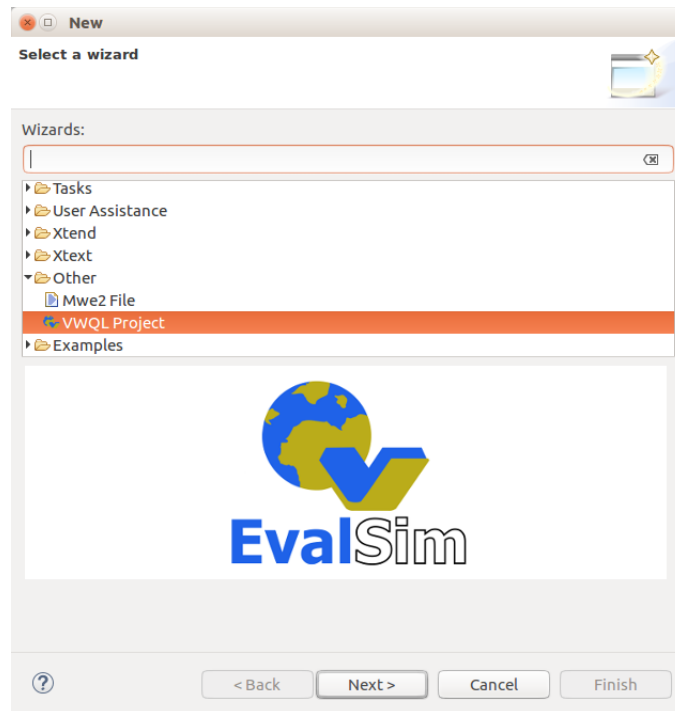


Figura A.1: Selección proyecto EvalSim o VWQL.

Cuando esté seleccionado pulsamos *Siguiente* y podremos darle un nombre al proyecto y si queremos cambiar la localización donde se va a guardar el proyecto (Figura A.2).

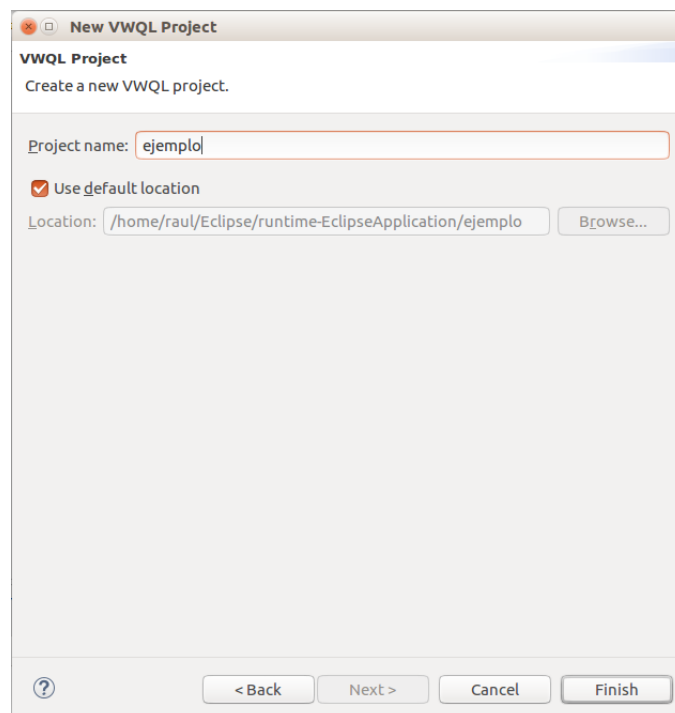


Figura A.2: Jerarquía de componentes Ecore.

Al pulsar *Finalizar*, se generará el proyecto EvalSim con la estructura que vemos en la Figura A.3. Como podemos observar, solo existe un archivo con extensión *.vwql*, que contiene una pequeña guía con la sintaxis del lenguaje y un ejemplo.



Figura A.3: Jerarquía de componentes Ecore.

Vamos al editor de EvalSim y modificamos la consulta existente, escribiendo el indicador que deseamos extraer.

La sintaxis del lenguaje VWQL se muestra en el Código A.1.

```

1 Evidence ( "evidence_name" ) :
2   ( | get ) students ( all | "student_name1", "student_name2", ... )
3   show ( game | words | words_german | sentences | single | turns | time | points
4         | greetings | checks | WHquestions | prepositions | confirmations
5         | communication_strategies | emojis
6         | ( distance ) ONLY IN HIDDENROOM GAME
7         | ( panel | panel_points ) ONLY IN MEMORY GAMES
8         | ( distance | client_points ) ONLY IN COFFEESHOP GAME
9   )
10  in ( all | Memory_Saturn | Memory_HiddenRoom | HiddenRoom | Saturn | Coffeeshop )
11  between ( yyyy/mm/dd ) and ( yyyy/mm/dd )
12  .

```

Código A.1: Lenguaje VWQL completo

Cuando tengamos la consulta escrita, sobre el editor abrimos el menú desplegable del botón secundario del ratón, y seleccionamos la opción *Generate Code* (Figura A.4).

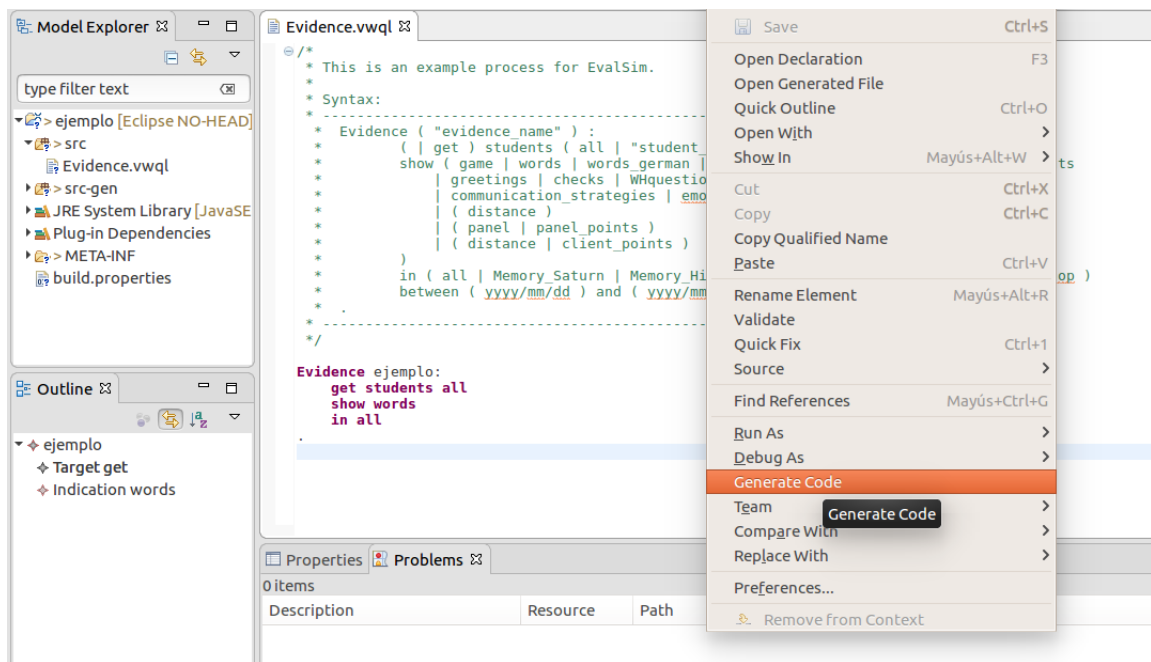


Figura A.4: Jerarquía de componentes Ecore.

El proceso puede tardar algunos minutos dependiendo de la consulta realizada. Al finalizar obtendremos la estructura que se muestra en la Figura A.1.

Si no aparece la carpeta *output*, tenemos que pinchar con el ratón en la carpeta del proyecto y dar al botón F5 del teclado; o podemos desplegar el menú del botón secundario del ratón, sobre la carpeta del proyecto, seleccionando la opción *Actualizar*.

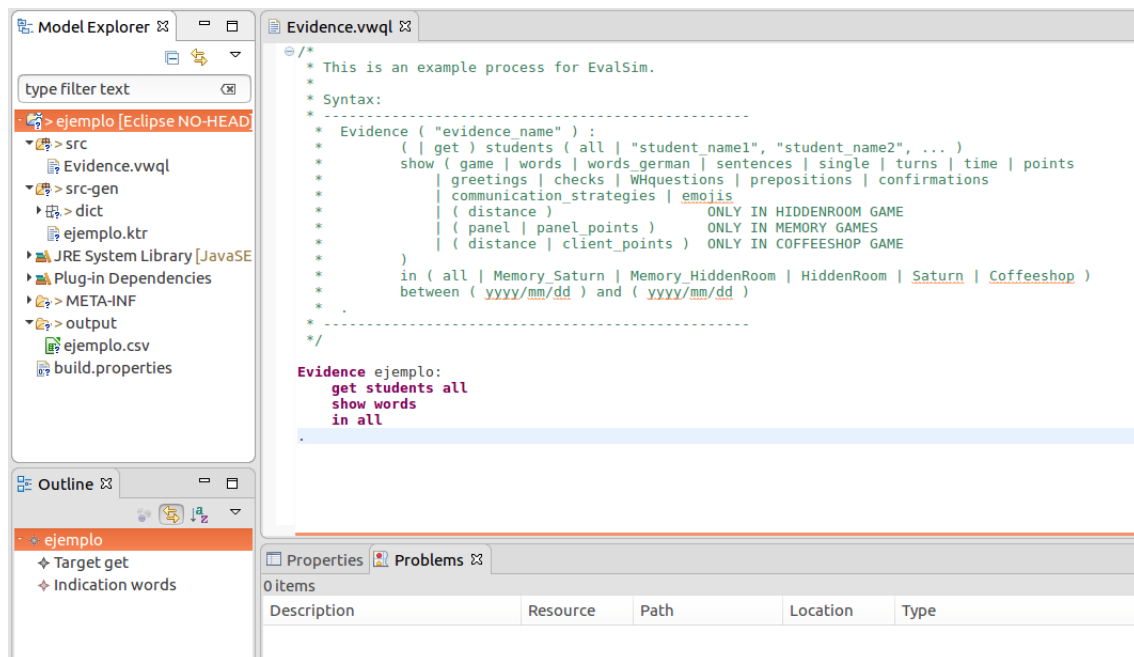


Figura A.5: Jerarquía de componentes Ecore.

En la carpeta *output* podremos encontrar el fichero resultado de la consulta (Figura A.6). Este fichero se llamará exactamente igual que el nombre que hemos dado a la *Evidencia* pero con extensión *.csv*.

Generalmente al pinchar dos veces sobre el fichero, se abrirá con algún software de hojas de cálculo, o si no tenemos instalado ninguno, con cualquier editor de texto. Si contamos con software de hoja de cálculo, debemos seleccionar como separador punto y coma.

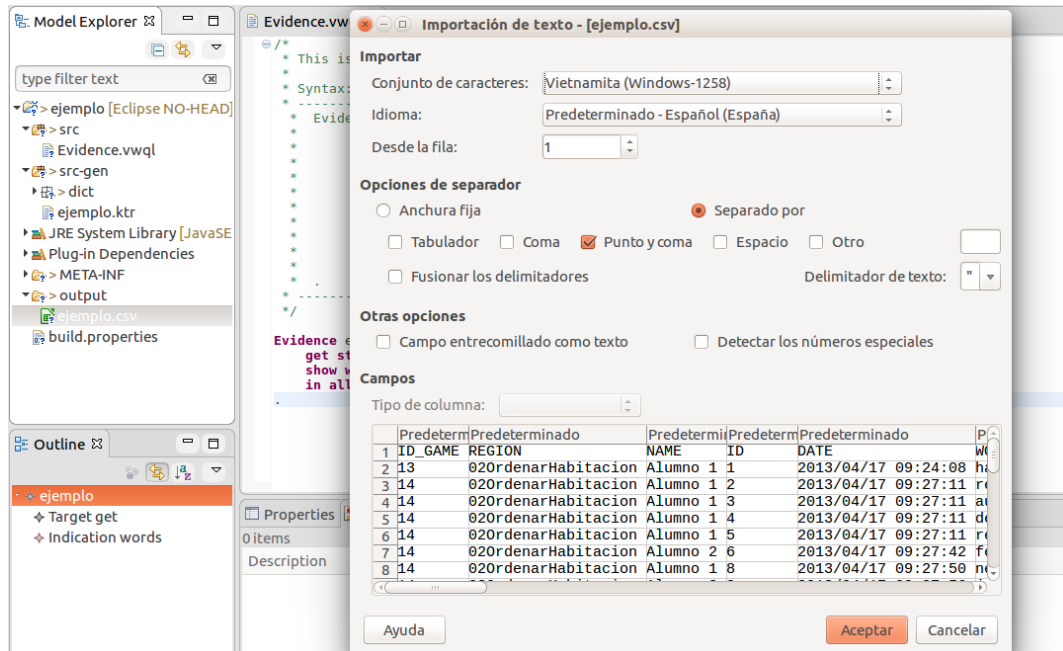


Figura A.6: Jerarquía de componentes Ecore.

En la Figura A.7 podemos ver el resultado de la consulta que hemos usado de ejemplo.

	A	B	C	D	E	F	
1	ID_GAME	REGION	NAME	ID	DATE	WORD	MESSAGE
2	13	02OrdenarHabitacion	Alumno 1	1	2013/04/17 09:24:08	hallo	Hallo!
3	14	02OrdenarHabitacion	Alumno 1	2	2013/04/17 09:27:11	rechts	rechts auf dem regal
4	14	02OrdenarHabitacion	Alumno 1	3	2013/04/17 09:27:11	auf	rechts auf dem regal
5	14	02OrdenarHabitacion	Alumno 1	4	2013/04/17 09:27:11	dem	rechts auf dem regal
6	14	02OrdenarHabitacion	Alumno 1	5	2013/04/17 09:27:11	regal	rechts auf dem regal
7	14	02OrdenarHabitacion	Alumno 2	6	2013/04/17 09:27:42	fertig	fertig!
8	14	02OrdenarHabitacion	Alumno 1	8	2013/04/17 09:27:50	nein	nein
9	14	02OrdenarHabitacion	Alumno 2	9	2013/04/17 09:27:56	jaaa	Jaaa
10	14	02OrdenarHabitacion	Alumno 1	10	2013/04/17 09:28:00	unter	unter
11	14	02OrdenarHabitacion	Alumno 2	11	2013/04/17 09:28:12	unter?	unter? oder auf!???
12	14	02OrdenarHabitacion	Alumno 2	12	2013/04/17 09:28:12	oder	unter? oder auf!???
13	14	02OrdenarHabitacion	Alumno 2	13	2013/04/17 09:28:12	auf	unter? oder auf!???
14	14	02OrdenarHabitacion	Alumno 1	14	2013/04/17 09:28:21	auf	auf aber unter
15	14	02OrdenarHabitacion	Alumno 1	15	2013/04/17 09:28:21	aber	auf aber unter
16	14	02OrdenarHabitacion	Alumno 1	16	2013/04/17 09:28:21	unter	auf aber unter
17	14	02OrdenarHabitacion	Alumno 1	17	2013/04/17 09:28:27	ist	ist korrekt aber
18	14	02OrdenarHabitacion	Alumno 1	18	2013/04/17 09:28:27	korrekt	ist korrekt aber
19	14	02OrdenarHabitacion	Alumno 1	19	2013/04/17 09:28:27	aber	ist korrekt aber
20	14	02OrdenarHabitacion	Alumno 2	20	2013/04/17 09:28:39	fertig	fertig?
21	14	02OrdenarHabitacion	Alumno 1	21	2013/04/17 09:28:42	nein	nein
22	14	02OrdenarHabitacion	Alumno 2	22	2013/04/17 09:28:43	korrekt	korrekt?
23	14	02OrdenarHabitacion	Alumno 2	24	2013/04/17 09:28:51	was	:(was ist los?
24	14	02OrdenarHabitacion	Alumno 2	25	2013/04/17 09:28:51	ist	:(was ist los?
25	14	02OrdenarHabitacion	Alumno 2	26	2013/04/17 09:28:51	los	:(was ist los?
26	14	02OrdenarHabitacion	Alumno 1	27	2013/04/17 09:28:52	auf	auf dem regal
27	14	02OrdenarHabitacion	Alumno 1	28	2013/04/17 09:28:52	dem	auf dem regal
28	14	02OrdenarHabitacion	Alumno 1	29	2013/04/17 09:28:52	regal	auf dem regal
29	14	02OrdenarHabitacion	Alumno 2	30	2013/04/17 09:29:23	ich	ich verstehen nicht!!!!
30	14	02OrdenarHabitacion	Alumno 2	31	2013/04/17 09:29:23	verstehen	ich verstehen nicht!!!!
31	14	02OrdenarHabitacion	Alumno 2	32	2013/04/17 09:29:23	nicht	ich verstehen nicht!!!!
32	14	02OrdenarHabitacion	Alumno 2	33	2013/04/17 09:29:31	mist	mist! :/

Figura A.7: Jerarquía de componentes Ecore.

Apéndice B

Manual de Instalación EvalSim

En este manual se detalla el proceso de instalación y explotación del sistema EvalSim.

B.1. Requisitos previos

Para la instalación de EvalSim necesita los siguientes requisitos:

- Eclipse, con la herramienta EMF.
- Pentaho (necesario para ejecutar las transformaciones fuera de EvalSim).

B.2. Procedimientos de instalación

Si no dispone de Eclipse, lo puede descargar en:

- Eclipse Modeling: <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/neon3>

Si dispone de Eclipse instalado:

- Descargar los plugins de EvalSim:
 - Plugins EvalSim: <https://bitbucket.org/RaulGS/vwql/downloads/EvalSim-plugins.zip>
- Copiar los plugins de EvalSim en la carpeta plugins, ubicada en la carpeta raíz de Eclipse (Figura B.1).
- Ya puede ejecutar Eclipse y seleccionar un proyecto de tipo *VWQL Project*.

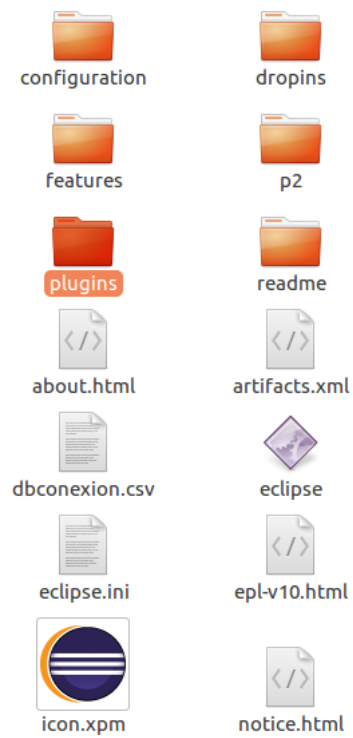


Figura B.1: Carpeta plugins dentro de carpeta Eclipse.

Apéndice C

Bases de Datos

C.1. Base de Datos: Avatar Tracking

La base de datos *avatar-tracking* (Figura C.1) registra en la tabla *anke* la posición de un avatar en un determinado instante. Para ello el avatar debe colocarse una pulsera en el mundo virtual, que cada 10 segundos envía la localización exacta del avatar a esta BD.

anke
+name: VARCHAR
+uuid: VARCHAR
+region: VARCHAR
+position_x: FLOAT
+position_y: FLOAT
+position_z: FLOAT
+rotation: FLOAT
+timestamp: TIMESTAMP

Figura C.1: Diagrama ER de la base de datos *avatar-tracking*.

C.1.1. Tabla *anke*

Descripción del contenido de la tabla *anke* de la BD *avatar-tracking* (Tabla C.1).

CAMPO	DESCRIPCIÓN
name	Nombre del avatar que envía la información.
uuid	Identificador único del avatar que envía la información.
region	Nombre de la región en la que se encuentra el avatar.
position_x	Coordenada X de la región en la que se encuentra el avatar.
position_y	Coordenada Y de la región en la que se encuentra el avatar.
position_z	Coordenada Z de la región en la que se encuentra el avatar.
rotation	Rotación del avatar respecto de la región en la que se encuentra.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos.

Tabla C.1: Tabla *anke* de la BD *avatar-tracking*.

C.2. Base de Datos: UCA Cafeteria Version 2

La base de datos *uca_cafeteriaversion2* (Figura C.2) almacena todos datos relativos al juego de la cafeteria, en el que compiten dos alumnos por ver cuantas mesas son capaces de servir en 15 minutos. Se almacenan los nombres de los jugadores, los puntos conseguidos, y los clientes que ha servido y los puntos que ha conseguido por cada cliente.

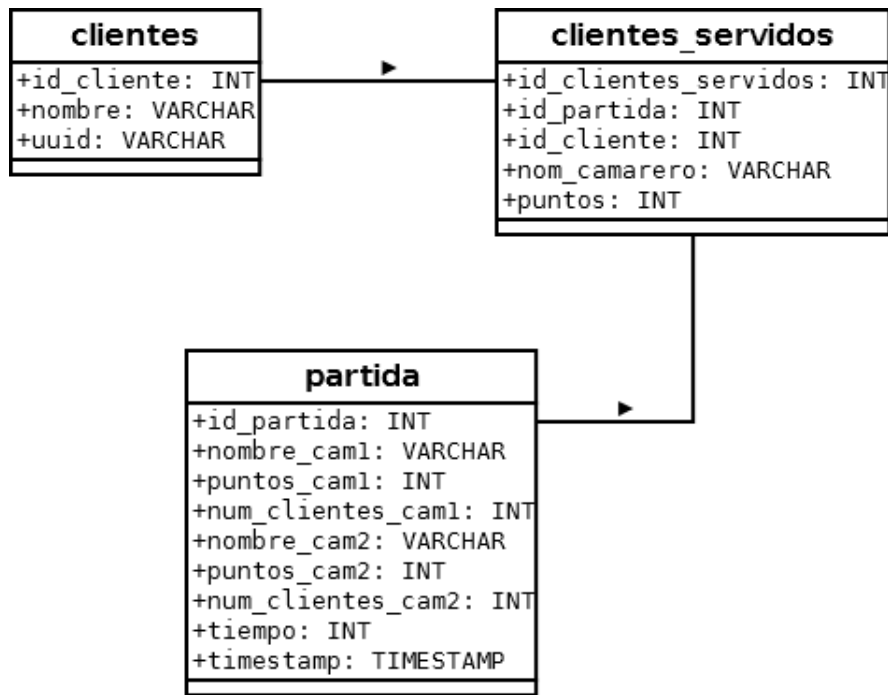


Figura C.2: Diagrama ER de la base de datos *uca_cafeteriaversion2*.

C.2.1. Tabla *clientes*

Descripción del contenido de la tabla *clientes* de la BD *uca_cafeteriaversion2* (Tabla C.2). En esta tabla se almacena los datos de los clientes que existen, es decir, el nombre y uuid de los bots de este juego.

CAMPO	DESCRIPCIÓN
id_cliente	Identificador del cliente/bot en esta base de datos.
nombre	Nombre del cliente/bot.
uuid	Identificador único en el mundo virtual del cliente/bot.

Tabla C.2: Tabla *clientes* de la BD *uca_cafeteriaversion2*.

C.2.2. Tabla *clientes_servidos*

Descripción del contenido de la tabla *clientes_servidos* de la BD *uca_cafeteriaversion2* (Tabla C.3). En esta tabla se almacenan los puntos que un jugador ha ido consiguiendo de cada cliente a lo largo de cada partida.

CAMPO	DESCRIPCIÓN
id_clientes_servidos	Identificador de la tabla.

id_partida	Identificador de la partida en la que ocurrió la instancia.
id_cliente	Identificador del cliente/bot al que sirvió el jugador.
nom_camarero	Nombre del jugador que realizó la acción.
puntos	Puntuación que consiguió el jugador.

Tabla C.3: Tabla *clientes_servidos* de la BD *uca_cafeteriaversion2*.

C.2.3. Tabla *partida*

Descripción del contenido de la tabla *partida* de la BD *uca_cafeteriaversion2* (Tabla C.4). En esta tabla se almacenan los puntos totales que los jugadores han conseguido en una partida y el número de clientes/bots que han servido en una determinada partida.

CAMPO	DESCRIPCIÓN
id_partida	Identificador de la partida.
nombre_cam1	Nombre que identifica al jugador 1.
puntos_cam1	Puntuación que consiguió el jugador 1 en la partida.
num_clientes_cam1	Número de clientes que sirvió el jugador 1 en la partida.
nombre_cam2	Nombre que identifica al jugador 2.
puntos_cam2	Puntuación que consiguió el jugador 2 en la partida.
num_clientes_cam2	Número de clientes que sirvió el jugador 2 en la partida.
tiempo	Tiempo en segundos que duró la partida.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora de finalización de la partida.

Tabla C.4: Tabla *partida* de la BD *uca_cafeteriaversion2*.

C.3. Base de Datos: UCA Memory Ordenar Habitación

La base de datos *uca_memory_ordenar_habitacion* (Figura C.3) almacena todos datos relativos a partidas de los juegos memory con vocabulario de objetos y situaciones de esos objetos, y ordenar la habitación. En el primer juego los alumnos interactúan de forma individual, por lo que solo se registra un jugador por partida. El segundo juego se realiza de forma cooperativa por parejas, de forma que se almacena la puntuación conjunta de la pareja.

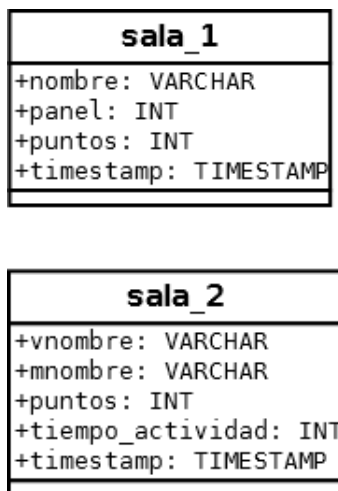


Figura C.3: Diagrama ER de la base de datos *uca_memory_ordenar_habitacion*.

C.3.1. Tabla *sala_1*

Descripción del contenido de la tabla *sala_1* de la BD *uca_memory_ordenar_habitacion* (Tabla C.5). En esta tabla se almacenan los datos del juego memory.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador.
panel	Número del panel al que se ha jugado.
puntos	Puntuación que consiguió el jugador en la partida.
tiempo	Tiempo que tarda un jugador en acabar la partida en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora de finalización de la partida.

Tabla C.5: Tabla *sala_1* de la BD *uca_memory_ordenar_habitacion*.

C.3.2. Tabla *sala_2*

Descripción del contenido de la tabla *sala_2* de la BD *uca_memory_ordenar_habitacion* (Tabla C.6). En esta tabla se almacenan los datos del juego *ordenar la habitación*.

CAMPO	DESCRIPCIÓN
vnombre	Nombre que identifica al jugador que lleva el visor de imágenes.
mnombre	Nombre que identifica al jugador que lleva el mando.

puntos	Puntuación que consiguieron ambos jugadores en la partida de forma conjunta.
tiempo_actividad	Tiempo que tardó la pareja de jugadores en acabar la partida en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora de finalización de la partida.

Tabla C.6: Tabla *sala_2* de la BD *uca_memory_ordenar_habitacion*.

C.4. Base de Datos: UCA Coffeeshop

La base de datos *uca_coffeeshop* (Figura C.4) almacena todos datos relativos a los juegos:

- Cafetería, juego de rol en el que un jugador debe actuar como un camarero y servir mesas.
- Casino, en el que un jugador debe emparejar audios, textos e imágenes, de varias ruletas, según corresponda.
- Mango, juego de rol en el que un jugador debe actuar como cliente y pedir al dependiente ciertos artículos de ropa.
- Quiz, los jugadores se enfrentarán y tendrán que reconocer la palabra que se asocia a la imagen que se les muestra, escribiéndola en el chat en un tiempo determinado.
- Saturn, juego de rol en el que dos jugadores deben comprar en una tienda. Uno lleva el carro de la compra y el otro la lista de la compra.

De todos estos juegos se almacenan los nombres de los jugadores, los puntos conseguidos, el tiempo que han tardado en realizar la actividad y algún dato adicional más, específico del juego.

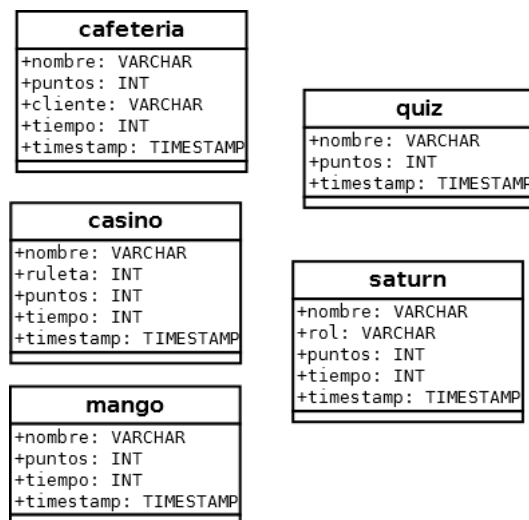


Figura C.4: Diagrama ER de la base de datos *uca_coffeeshop*.

C.4.1. Tabla *cafeteria*

Descripción del contenido de la tabla *cafeteria* de la BD *uca_coffeeshop* (Tabla C.7). En esta tabla se almacenan los datos del juego *cafeteria*.

En este juego el jugador, que actúa de camarero, debe servir varias mesas ocupadas por clientes (bots) que se encuentran repartidos por una cafetería. Cada instancia de la tabla representa a un cliente que el jugador ha servido.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que realiza el rol de camarero.
puntos	Puntuación que consiguió el jugador al servir al cliente.
cliente	Nombre que identifica al cliente/bot que fue servido.

tiempo	Tiempo que tardó el jugador en servir al cliente en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador terminó de servir al cliente.

Tabla C.7: Tabla *cafeteria* de la BD *uca_coffeeshop*.

C.4.2. Tabla *casino*

Descripción del contenido de la tabla *casino* de la BD *uca_coffeeshop* (Tabla C.8). En esta tabla se almacenan los datos del juego *casino*.

En este juego el jugador tendrá que completar un total de seis ruletas, emparejando audios, texto e imágenes según corresponda. Cada instancia de la tabla representa que un jugador ha completado una ruleta en el menor tiempo posible.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que juega a la ruleta.
ruleta	Identificador de la ruleta que a la que se ha jugado.
puntos	Puntuación que consiguió el jugador al completar la ruleta.
tiempo	Tiempo que tardó el jugador en terminar el juego.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador terminó la ruleta.

Tabla C.8: Tabla *casino* de la BD *uca_coffeeshop*.

C.4.3. Tabla *mango*

Descripción del contenido de la tabla *mango* de la BD *uca_coffeeshop* (Tabla C.9). En esta tabla se almacenan los datos del juego *mango*.

En este juego el jugador, que actúa de cliente en una tienda de ropa, debe pedir a través del chat al dependiente de la tienda (bot) los artículos de una lista de la compra. Cada instancia de la tabla representa una partida del jugador.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que realiza el rol de cliente de la tienda.
puntos	Puntuación que consiguió el jugador al adquirir los artículos.
tiempo	Tiempo que tardó el jugador en adquirir los artículos en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador terminó de realizar la compra.

Tabla C.9: Tabla *mango* de la BD *uca_coffeeshop*.

C.4.4. Tabla *quiz*

Descripción del contenido de la tabla *quiz* de la BD *uca_coffeeshop* (Tabla C.10). En esta tabla se almacenan los datos del juego *quiz*.

En este juego varios jugadores se enfrentarán y tendrán que reconocer la palabra que se asocia a la

imagen que se les muestra, escribiéndola en el chat en un tiempo determinado. Cada instancia de la tabla representa una partida de un jugador.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que realiza el quiz.
puntos	Puntuación que consiguió el jugador en la partida de quiz.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador terminó de realizar la actividad.

Tabla C.10: Tabla *quiz* de la BD *uca_coffeeshop*.

C.4.5. Tabla *saturn*

Descripción del contenido de la tabla *saturn* de la BD *uca_coffeeshop* (Tabla C.11). En esta tabla se almacenan los datos del juego *saturn*.

En este juego por pareja, los jugadores actúan como clientes, pero cada uno desempeña un rol. Uno de los jugadores lleva el carrito de la compra, mientras que el otro tiene una lista de la compra. Ambos jugadores deben comprar los artículos que se les indican en varias listas en el menor tiempo posible. La comunicación entre ambos se realiza a través del chat. Cada instancia de la tabla representa una partida de uno de los jugadores, indicando el rol que había desempeñado en la partida.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que realiza compras en la tienda.
rol	Identifica si el rol que había desempeñado en la partida era portador del carro o de la lista de la compra.
puntos	Puntuación conjunta que consiguió el jugador al adquirir los artículos.
tiempo	Tiempo conjunto que tardó el jugador en adquirir los artículos en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que los jugadores terminaron de realizar la compra.

Tabla C.11: Tabla *saturn* de la BD *uca_coffeeshop*.

C.5. Base de Datos: UCA Memory Saturn

La base de datos *uca_memory_saturn* (Figura C.5) almacena todos datos relativos a partidas del juego memory con vocabulario de objetos y situaciones de esos objetos relativos a electrodomesticos de la tienda *Saturn*. En este juego los alumnos interactúan de forma individual, por lo que solo se registra un jugador por partida.

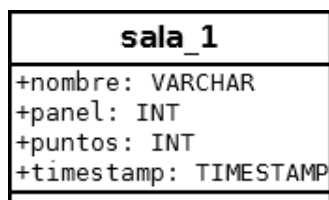


Figura C.5: Diagrama ER de la base de datos *uca_memory_saturn*.

C.5.1. Tabla *sala_1*

Descripción del contenido de la tabla *sala_1* de la BD *uca_memory_saturn* (Tabla C.12). En esta tabla se almacenan los datos del juego memory.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador.
panel	Número del panel al que se ha jugado.
puntos	Puntuación que consiguió el jugador en la partida.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora de finalización de la partida.

Tabla C.12: Tabla *sala_1* de la BD *uca_memory_saturn*.

C.6. Base de Datos: UCA Supermercado

La base de datos *uca_supermercado* (Figura C.6) almacena todos datos relativos a los juegos:

- Aktivitat 2, es un juego en el que el jugador escuchará un sonido y tendrá que buscar la imagen correspondiente.
- Aktivitat 3, en el que un jugador debe unir audios, textos e imágenes, de varias ruletas, según corresponda.
- Aktivitat 4, juego en el que el alumno usa todos los conocimientos adquiridos para completar varias listas de la compra buscando los productos en el supermercado.

De todos estos juegos se almacenan los nombres de los jugadores, los puntos conseguidos, el tiempo que han tardado en realizar la actividad y algún dato adicional más, específico del juego.



Figura C.6: Diagrama ER de la base de datos *uca_supermercado*.

C.6.1. Tabla *aktivitat2*

Descripción del contenido de la tabla *aktivitat2* de la BD *uca_supermercado* (Tabla C.13). En esta tabla se almacenan los datos del juego *Aktivitat 2* donde se enlazan sonidos con su imagen correspondiente.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador.
puntos	Puntuación que consiguió el jugador en la partida.
tiempo	Tiempo que tardó el jugador en completar la partida en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora de finalización de la partida.

Tabla C.13: Tabla *aktivitat2* de la BD *uca_supermercado*.

C.6.2. Tabla *aktivitat3*

Descripción del contenido de la tabla *aktivitat3* de la BD *uca_supermercado* (Tabla C.14). En esta tabla se almacenan los datos del juego *Aktivitat 3* donde un jugador debe unir audios, textos e imágenes, de varias ruletas, según corresponda.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que juega a la ruleta.
puntos	Puntuación que consiguió el jugador al completar la ruleta.
tiempo	Tiempo que tardó el jugador en terminar el juego.
ruleta	Identificador de la ruleta que a la que se ha jugado.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador terminó la ruleta.

Tabla C.14: Tabla *aktivitat3* de la BD *uca_supermercado*.

C.6.3. Tabla *aktivitat4*

Descripción del contenido de la tabla *aktivitat4* de la BD *uca_supermercado* (Tabla C.15). En esta tabla se almacenan los datos del juego *Aktivitat 4*.

En este juego de rol individual, el jugador actúa como cliente. Dicho jugador lleva un carrito y una lista de la compra. El jugador debe adquirir los artículos que se les indican en varias listas en el menor tiempo posible. Cada instancia de la tabla representa una partida.

CAMPO	DESCRIPCIÓN
nombre	Nombre que identifica al jugador que realiza compras en la tienda.
puntos	Puntuación que consiguió el jugador al adquirir los artículos.
tiempo	Tiempo que tardó el jugador en adquirir los artículos en segundos.
timestamp	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos, y que coincide con la hora en que el jugador de realizar la compra.

Tabla C.15: Tabla *aktivitat4* de la BD *uca_supermercado*.

C.7. Base de Datos: ROBUST

La siguiente base de datos que se describe en este apartado es creada de forma automática por el proceso de OpenSim que gestiona el grid privado (Figura C.7). Entendemos grid como un sistema de computación distribuido que permite compartir recursos no centrados geográficamente. Administra, entre otros factores, las posiciones de cada región en el mundo virtual, el inventario de los avatares y el control de acceso al mundo virtual.

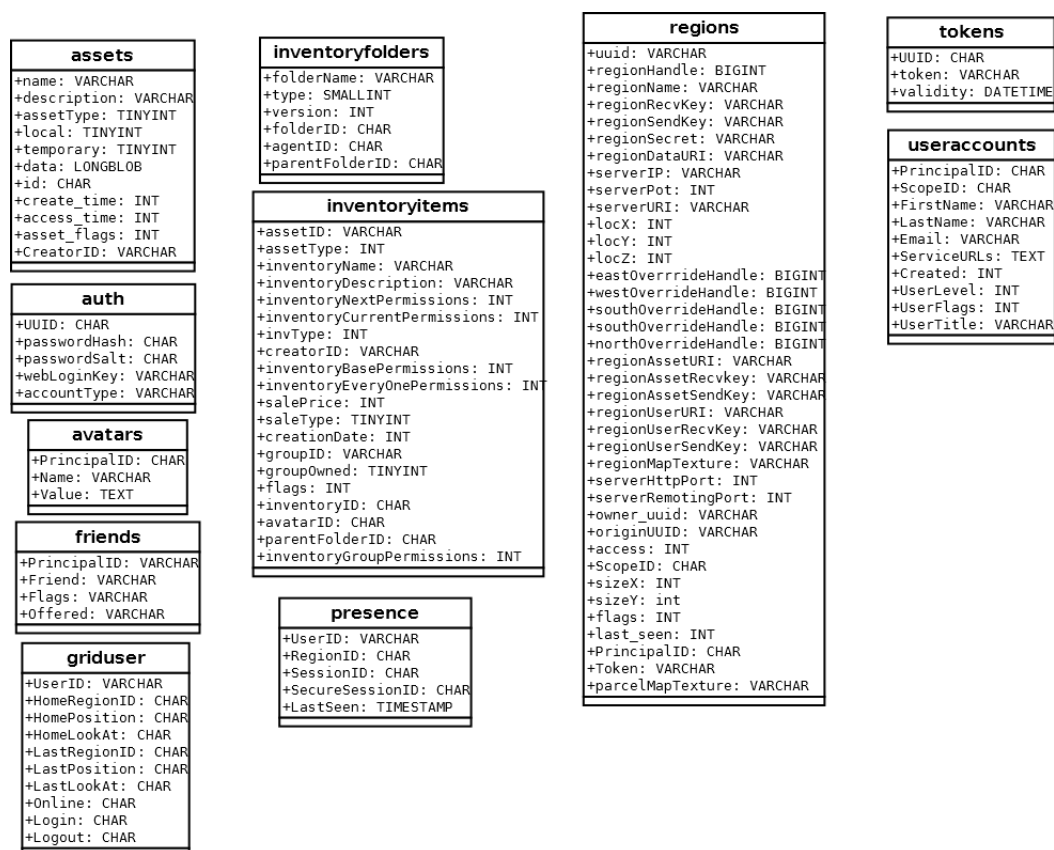


Figura C.7: Diagrama ER de la base de datos *robust*.

C.7.1. Tabla *assets*

Descripción del contenido de la tabla *assets* de la BD *robust* (Tabla C.16). En esta tabla se almacenan los objetos y elementos activos del grid como son formas geométricas (prims), ropa, texturas, notas, scripts, cuerpos (shape), pieles (skin) y otras piezas que usa el avatar para su arreglo personal o para construir edificios.

CAMPO	DESCRIPCIÓN
name	Nombre del elemento activo tal como aparecería en el inventario.
description	Campo de descripción de los elementos activos, tal como aparece en varios widgets de edición en el cliente.

assetType	Entero que designa el tipo de elemento activo. Los valores válidos son: <ul style="list-style-type: none"> ■ 0 - Texture in JPEG2000 J2C stream format ■ 1 - Sound ■ 2 - Calling Card ■ 3 - Landmark ■ 5 - Clothing ■ 6 - Object ■ 7 - Notecard ■ 10 - LSLText (también conocido como script) ■ 13 - Body Part ■ 20 - Animation ■ 21 - Gesture ■ 49 - Mesh
local	Para uso futuro.
temporary	Para uso futuro.
data	Este campo de tipo BLOB, contiene el script real, texto del notecard, imagen binaria (textura) u otros datos que se caracterizan por la columna Type.
id	UUID que identifica de forma inequívoca al elemento activo dentro del sistema.
create_time	Fecha y hora en que el elemento activo fue creado en formato estándar de UNIX.
access_time	Fecha y hora en que el elemento activo fue usado por última vez en formato estándar de UNIX.
asset_flags	Marcadores del elemento activo.
CreatorID	UUID del avatar que lo creó.

Tabla C.16: Tabla *assets* de la BD *robust*.

C.7.2. Tabla *auth*

Descripción del contenido de la tabla *auth* de la BD *robust* (Tabla C.17). Esta tabla contiene la información de autenticación de los usuarios. Lleva a cabo una versión cifrada de la contraseña utilizada por el usuario para iniciar sesión en OpenSim.

CAMPO	DESCRIPCIÓN
UUID	Identificador Universalmente único de un avatar.

passwordHash	Un resumen MD5 de la contraseña de los usuarios y de la cadena sal. El resumen se calcula de la siguiente manera: md5 (md5 (password) +  passwordSalt).
passwordSalt	Cadena usada en la encriptación de la contraseña.
webLoginKey	
accountType	Una cadena que puede usarse para distinguir diferentes tipos de cuentas de avatar.

Tabla C.17: Tabla *auth* de la BD *robust*.

C.7.3. Tabla *avatars*

Descripción del contenido de la tabla *avatars* de la BD *robust* (Tabla C.18). Esta tabla almacena información de la apariencia visual de los avatares y objetos adjuntos que éstos portan.

CAMPO	DESCRIPCIÓN
PrincipalID	Identificador Universalmente único de un avatar.
Name	<p>Nombre del objeto adjunto, archivo adjunto o propiedad.</p> <ul style="list-style-type: none"> ▪ Serial ▪ AvatarHeight - The height of the avatar (in meters) ▪ Legacy Wearables <ul style="list-style-type: none"> • BodyItem • SkinItem • HairItem • EyesItem • ShirtItem • PantsItem • ShoesItem • SocksItem • JacketItem • GlovesItem • UnderShirtItem • UnderPantsItem • SkirtItem • VisualParams ▪ New Style Wearables <ul style="list-style-type: none"> • AvatarType • Wearable n:m

	<p>Puede haber múltiples "Wearable n:m" líneas en la tabla. El valor de n indica para que parte del cuerpo es el objeto usable. El valor de m normalmente es 0. Habrá múltiples registros con el mismo valor de n, pero diferentes valores de m, si hay varios anexos a la misma parte del cuerpo. Los valores más altos de m representan objetos usables que están en la cima de los objetos usables con menores valores de m.</p> <p>Los valores válidos de n se muestran a continuación:</p> <ul style="list-style-type: none"> ■ 0 for BODY ■ 1 for SKIN ■ 2 for HAIR ■ 3 for EYES ■ 4 for SHIRT ■ 5 for PANTS ■ 6 for SHOES ■ 7 for SOCKS ■ 8 for JACKET ■ 9 for GLOVES ■ 10 for UNDERSHIRT ■ 11 for UNDERPANTS ■ 12 for SKIRT ■ 13 for ALPHA ■ 14 for TATTOO
Value	Los datos específicos de la propiedad del avatar mencionado.

Tabla C.18: Tabla *avatars* de la BD *robust*.

C.7.4. Tabla *friends*

Descripción del contenido de la tabla *friends* de la BD *robust* (Tabla C.19). Esta tabla contiene información de los amigos de los usuarios.

CAMPO	DESCRIPCIÓN
PrincipalID	Identificador Universalmente único de un avatar.
Friend	Identificador Universalmente único de un avatar distinto del principal.
Flags	Marcadores de la tabla.
Offered	

Tabla C.19: Tabla *friends* de la BD *robust*.

C.7.5. Tabla *griduser*

Descripción del contenido de la tabla *griduser* de la BD *robust* (Tabla C.20). Esta tabla contiene información sobre el usuario en esta red en particular, para todos los que usan la red, local y extranjera. Esto incluye su región de origen, la última región, el estado en línea y la última de conexión/desconexión. Esta información es distinta de la información de autenticación.

CAMPO	DESCRIPCIÓN
UserID	Identificador Universalmente único del avatar.
HomeRegionID	UUID de la región de origen.
HomePosition	Coordenadas de la posición de origen.
HomeLookAt	Orientación de origen.
LastRegionID	UUID de la última región en la que estuvo el avatar.
LastPosition	Coordenadas de la última posición en la que estaba el avatar.
LastLookAt	Orientación de la ultima vez que se desconecto el avatar.
Online	Estado en línea del avatar.
Login	última conexión del avatar.
Logout	última desconexión del avatar.

Tabla C.20: Tabla *griduser* de la BD *robust*.

C.7.6. Tabla *inventoryfolder*

Descripción del contenido de la tabla *inventoryfolder* de la BD *robust* (Tabla C.21). En esta tabla se registran los detalles relativos a la organización del inventario del avatar, concretamente las carpetas. No hay información sobre el contenido de las carpetas es en esta tabla.

CAMPO	DESCRIPCIÓN
folderName	Texto con el nombre de la carpeta tal y como aparece en la interfaz del inventario del cliente.
type	<p>Tipo de la carpeta del inventario. Esto es un subconjunto del tipo enumerado InventoryType solo para carpetas.</p> <ul style="list-style-type: none"> ■ -1 User Defined ■ 0 Textures ■ 1 Sounds ■ 2 Calling Cards ■ 3 Landmarks ■ 5 Clothing ■ 6 Objects ■ 7 Notecards

	<ul style="list-style-type: none"> ■ 8 My Inventory (OpenSimulator 0.6.x) ■ 9 My Inventory (OpenSimulator 0.7.x) ■ 10 Scripts ■ 13 Body Parts ■ 14 Trash ■ 15 Photo Album ■ 16 Lost and Found ■ 20 Animations ■ 21 Gestures ■ 22 Meshes ■ 23 Favorites ■ 47 Outfits ■ 100 My Suitcase
version	Este artículo se incrementa cuando un artículo es añadido a la carpeta.
folderID	Identifica unívocamente la carpeta. Esto facilita la carpeta que se reclama como un contenedor lógico por otras carpetas o las cosas en la tabla <i>inventoryitems</i> que se dice que están "dentro de" la carpeta.
agentID	Identificador de agente único. Identifica al agente que es propietario de la carpeta.
parentFolderID	Identifica la carpeta principal que contiene esta carpeta. Por lo tanto, debe coincidir con otra entrada de la tabla <i>inventoryfolder</i> que tiene que <i>folderID</i> .

Tabla C.21: Tabla *inventoryfolder* de la BD *robust*.

C.7.7. Tabla *inventoryitems*

Descripción del contenido de la tabla *inventoryitems* de la BD *robust* (Tabla C.22). Esta tabla contiene detalles de todos los artículos del inventario (incluyendo asociaciones de carpetas) exceptuando el contenido. Desde el artículo se referencia al contenido que está almacenado en la tabla *assets*.

CAMPO	DESCRIPCIÓN
assetID	Hace referencia a la fila relacionada en la tabla activos.
assetType	Tipo de elemento activo (script, textura, etc.) .
inventoryName	Texto del nombre del artículo del inventario tal como aparece en la interfaz del inventario del cliente.

inventoryDescription	Texto de la descripción del artículo del inventario tal y como aparece en la interfaz del inventario del cliente.
inventoryNextPermissions	Los permisos que se aplicarán para el artículo cuando es transferido a otro avatar.
inventoryCurrentPermi. . .	Los permisos del artículo para el propietario actual del elemento.
invType	<p>Tipo del artículo del inventario. Esto es un subconjunto del tipo enumerado <i>InventoryType</i> solo para artículos.</p> <ul style="list-style-type: none"> ■ -1 User Defined ■ 0 Texture ■ 1 Sound ■ 2 Calling Card ■ 3 Landmark ■ 6 Object ■ 7 Notecard ■ 10 Script ■ 15 Snapshot ■ 17 Attachment ■ 18 Wearable (assets 5 and 13) ■ 19 Animation (assets 20) ■ 21 Gesture ■ 22 Mesh (asset 49)
creatorID	UUID del creador original del artículo.
inventoryBasePermissions	Los permisos base del objeto.
inventoryEveryOnePer. . .	Permiso que afecta a todos excepto al propietario del elemento.
salePrice	Precio de venta del artículo.
saleType	Tipo de venta del artículo.
creationDate	Fecha en la que el artículo fue creado.
groupID	UUID del grupo al que el artículo pertenece.
groupOwned	Marcadores de propiedad del grupo.
flags	Marcadores de propiedad del grupo.
inventoryID	UUID del artículo del inventario.
avatarID	UUID del propietario del artículo.
parentFolderID	UUID de la carpeta que contiene al artículo. Este ID de la carpeta debe coincidir con una entrada de la tabla <i>inventoryFolders</i> .
inventoryGroupPermiss. . .	Permisos con respecto al grupo de propietarios del artículo.

Tabla C.22: Tabla *inventoryitems* de la BD *robust*.

C.7.8. Tabla *presence*

Descripción del contenido de la tabla *presence* de la BD *robust* (Tabla C.23). Información de la región y de la sesión para los usuarios actualmente conectados. Se utiliza para el enrutamiento de mensajería instantánea entre otras cosas.

CAMPO	DESCRIPCIÓN
UserID	Identificador Universalmente único del avatar.
RegionID	Identificador de la región en la que se encuentra el avatar.
SessionID	Identificador de la sesión.
SecureSessionID	Identificador de seguridad de la sesión.
LastSeen	Hora de la última revisión.

Tabla C.23: Tabla *presence* de la BD *robust*.

C.7.9. Tabla *regions*

Descripción del contenido de la tabla *regions* de la BD *robust* (Tabla C.24). Esta tabla contiene información concerniente a todas las regiones actualmente conectadas en el grid.

CAMPO	DESCRIPCIÓN
uuid	Identificador Universalmente único de la región.
regionHandle	La dirección absoluta de las coordenadas de origen, se especifica en metros, de la región calculada a partir de LocX e como LocY ($\text{LocX} * 256 * 65536 + (\text{LocY} * 256)$).
regionName	El nombre de la región que aparece en el mapa. No tiene porque ser necesariamente único.
regionRecvKey	La contraseña usada por el servidor de la región para verificar la autenticidad de las comunicaciones de otras regiones.
regionSendKey	La contraseña usada por el servidor de la región en la verificación de su identidad en la comunicación con otras regiones.
regionSecret	
regionDataURI	
serverIP	Dirección IP del servidor de la región.
serverPort	Puerto de conexión para el servidor de la región.
serverURI	.
locX	Coordenada X de la región en el grid.
locY	Coordenada Y de la región en el grid.
locZ	Coordenada Z de la región en el grid.
eastOverrideHandle	.
westOverrideHandle	.
southOverrideHandle	.
northOverrideHandle	.
regionAssetURI	La URL y número del puerto del servidor del artículo para el grid.
regionAssetRecvKey	Contraseña usada para verificar la autenticidad de las comunicaciones recibidas del servidor del artículo.
regionAssetSendKey	Contraseña usada para verificar la autenticidad de las comunicaciones enviadas al servidor del artículo.
regionUserURI	La URL y el número del puerto del servidor del usuario en el grid.

regionUserRecvKey	Contraseña usada para verificar la autenticidad de las comunicaciones recibidas del servidor del usuario.
regionUserSendKey	Contraseña usada para verificar la autenticidad de las comunicaciones enviadas del servidor del usuario.
regionMapTexture	Textura para el mapa que es mostrado en el minimapa del cliente.
serverHttpPort	Puerto de la región que responderá a las respuestas de búsqueda HTTP.
serverRemotingPort	.
owner_uuid	UUID del avatar que es propietario de la región.
originUUID	Actualmente parece contener el UUID actual de la región.

Tabla C.24: Tabla *regions* de la BD *robust*.

C.7.10. Tabla *tokens*

Descripción del contenido de la tabla *tokens* de la BD *robust* (Tabla C.25). No está especificada información sobre esta tabla.

CAMPO	DESCRIPCIÓN
UUID	Identificador Universalmente único del token.
token	Identificador.
validity	Fecha y hora de validez del token.

Tabla C.25: Tabla *tokens* de la BD *robust*.

C.7.11. Tabla *useraccounts*

Descripción del contenido de la tabla *usseraccounts* de la BD *robust* (Tabla C.26). Esta tabla almacena información de la cuenta del usuario, como el nombre, nivel de usuario y el correo electrónico. Tan sólo de cuentas locales.

CAMPO	DESCRIPCIÓN
PrincipalID	Identificador Universalmente único del usuario.
ScopeID	.
FirstName	Nombre del usuario.
LastName	Apellidos del usuario.
Email	Correo electrónico del usuario.
ServiceURLs	.
Created	Fecha en la que fue creada la cuenta de usuario.
UserLevel	Nivel de la cuenta del usuario (indica entre otras cosas si la cuenta es de administrador).
UserFlags	Marcadores de la cuenta del usuario.
UserTitle	.

Tabla C.26: Tabla *useraccounts* de la BD *robust*.

C.8. Base de Datos: OPENSIM

La base de datos que se describe en este apartado es creada de forma automática por el proceso de OpenSim que gestiona cada región del mundo virtual (Figura C.8). Entendemos región como cada una de las parcelas cuadradas que componen el mundo virtual. Dentro de cada base de datos que compone una región nos podemos encontrar recursos tales como texturas, formas (objetos), audios... También se encuentra si es de día o de noche, la forma y textura que tiene el suelo, el agua, las nubes...

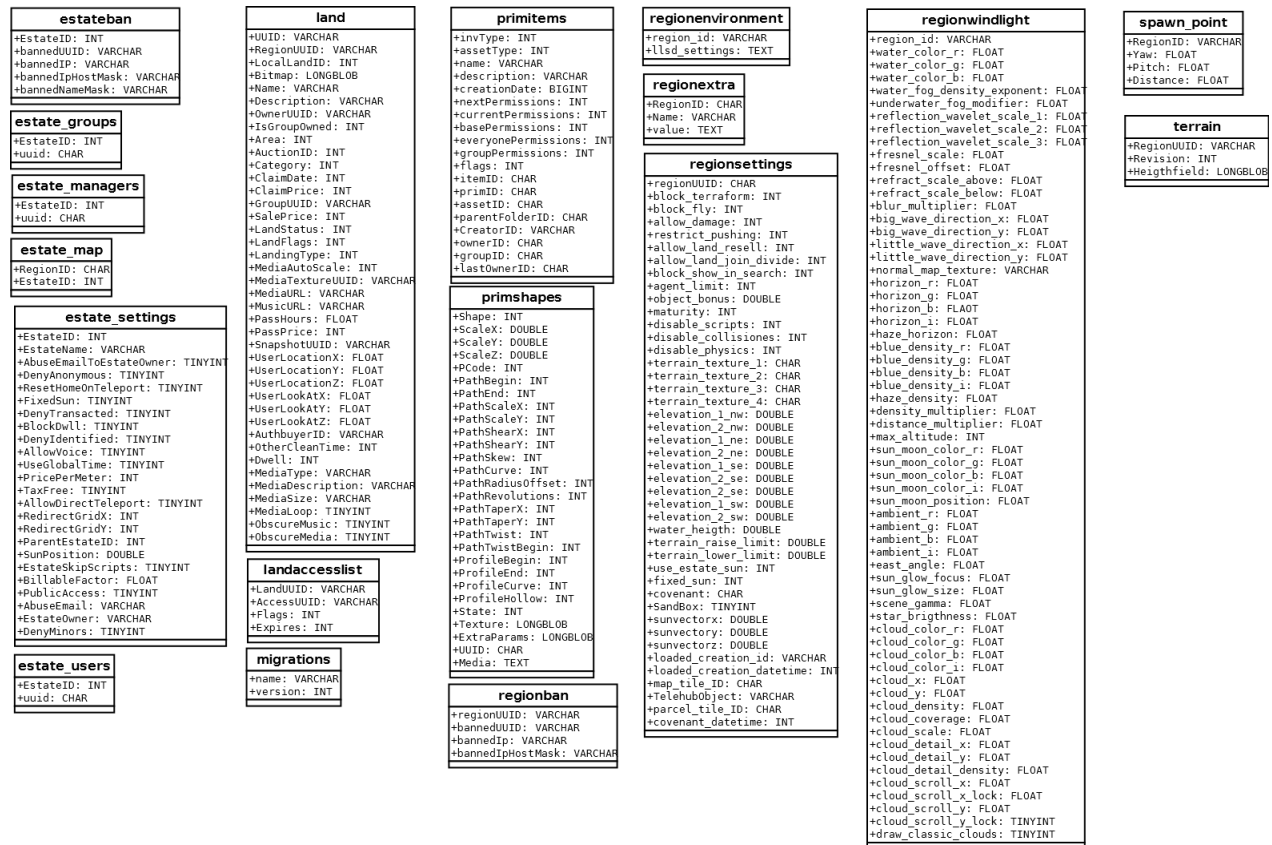


Figura C.8: Diagrama ER de la base de datos *os_*.

C.8.1. Tabla *estateban*

Descripción del contenido de la tabla *estateban* de la BD de *opensim*. En esta tabla se almacenan datos de los usuarios que tienen acceso restringido al Estado.

C.8.2. Tabla *estate_groups*

Descripción del contenido de la tabla *estate_groups* de la BD de *opensim*. En esta tabla se asocian los UUIDs de los grupos a un ID de Estado.

C.8.3. Tabla *estate_managers*

Descripción del contenido de la tabla *estate_managers* de la BD de *opensim*. En esta tabla se asocian los UUIDs de los administradores a un ID de Estado.

C.8.4. Tabla *estate_map*

Descripción del contenido de la tabla *estate_map* de la BD de *opensim*. En esta tabla se asocian los UUIDs de las regiones a un ID de Estado.

C.8.5. Tabla *estate_settings*

Descripción del contenido de la tabla *estate_settings* de la BD de *opensim*. En esta tabla se almacenan los datos de configuración de los Estados.

C.8.6. Tabla *estate_users*

Descripción del contenido de la tabla *estate_users* de la BD de *opensim*. En esta tabla se asocian los UUIDs de los usuarios que tienen acceso permitido al Estado.

C.8.7. Tabla *land*

Descripción del contenido de la tabla *land* de la BD de *opensim*. En esta tabla se almacenan los datos de las propiedades de las parcelas.

C.8.8. Tabla *landaccesslist*

Descripción del contenido de la tabla *landaccesslist* de la BD de *opensim*. En esta tabla se controla el acceso a las parcelas.

C.8.9. Tabla *migrations*

Descripción del contenido de la tabla *migrations* de la BD de *opensim*. En esta tabla se almacenan los datos relativos a la versión, de forma que se puedan realizar actualizaciones a nuevos formatos de bases de datos *primitems*.

C.8.10. Tabla *primitems*

Descripción del contenido de la tabla *primitems* de la BD de *opensim*. En esta tabla se almacenan datos sobre los elementos *Prim* y sus relaciones con *Usuarios* y con otros *Prims*.

C.8.11. Tabla *primshapes*

Descripción del contenido de la tabla *primshapes* de la BD de *opensim*. En esta tabla se almacenan la forma, color y textura de los *Prims*.

C.8.12. Tabla *regionban*

Descripción del contenido de la tabla *regionban* de la BD de *opensim*. En esta tabla se almacenan datos de los usuarios que tienen acceso restringido a la Región.

C.8.13. Tabla *regionenvironment*

Descripción del contenido de la tabla *regionenvironment* de la BD de *opensim*. En esta tabla se almacenan datos de configuración del entorno de la región.

C.8.14. Tabla *regionextra*

Descripción del contenido de la tabla *regionextra* de la BD de *opensim*. En esta tabla se almacenan el nombre y valores adicionales de las Regiones.

C.8.15. Tabla *regionsettings*

Descripción del contenido de la tabla *regionsettings* de la BD de *opensim*. En esta tabla se almacenan los datos de configuración de las Regiones.

C.8.16. Tabla *regionwindlight*

Descripción del contenido de la tabla *regionwindlight* de la BD de *opensim*. En esta tabla se almacenan los datos de configuración de viento y luz de las Regiones.

C.8.17. Tabla *spawn_point*

Descripción del contenido de la tabla *spawn_point* de la BD de *opensim*. En esta tabla se almacena el punto de aparición de las Regiones.

C.8.18. Tabla *terrain*

Descripción del contenido de la tabla *terrain* de la BD de *opensim*. En esta tabla se almacena el historial de revisión del terreno.

C.9. Base de Datos: OPENSIM MODIFICADO

En esta sección se describe la base de datos *os_modificada* (Tabla C.9). Esta base de datos es una mejora realizada por compañeros de la *Universidad Autónoma de Madrid* al propio código de *OpenSim* y a la base de datos que se genera automáticamente (es un anexo a la BD de la Figura C.8). Esta modificación nos permite almacenar datos tales como chat, eventos, acciones... que es producto de la interacción de los jugadores con el mundo virtual, y que aporta información útil para investigación docente, pero que *OpenSim* no almacena por si solo.

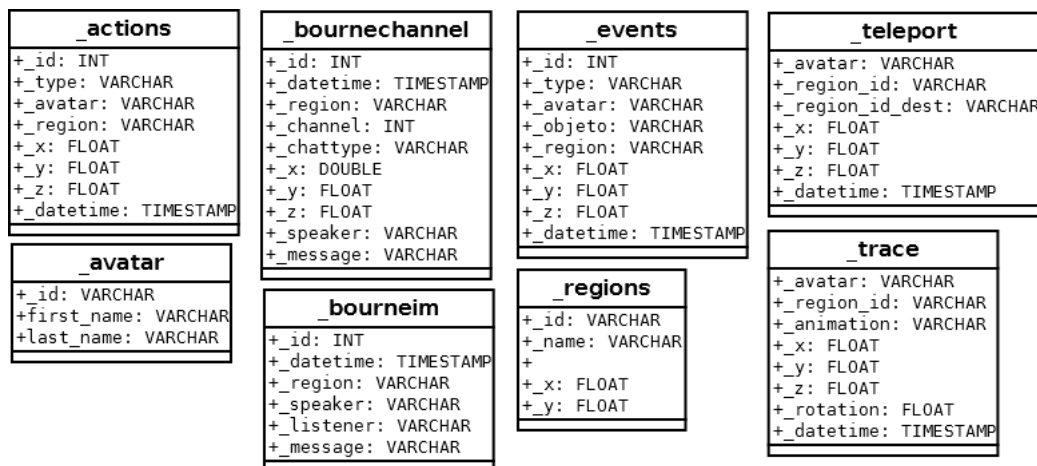


Figura C.9: Diagrama ER de la base de datos *os_modificada*.

C.9.1. Tabla *_actions*

Descripción del contenido de la tabla *_actions* de la BD *opensim* modificada (Tabla C.27). En esta tabla se almacenan registros de las acciones que pueden realizar los avatares dentro de *OpenSim*.

CAMPO	DESCRIPCIÓN
<i>_id</i>	Identificador de la instancia.
<i>_type</i>	Tipo de acción que realizó el avatar. Éstas son las acciones que quedan registradas. <ul style="list-style-type: none"> ▪ <i>_chat_privado</i> ▪ <i>_chat_publico</i> ▪ <i>_compile</i> ▪ <i>_compile_error</i> ▪ <i>_touch</i>
<i>_avatar</i>	UUID del avatar que realizó la acción.
<i>_region</i>	UUID de la región en la que se realizó la acción.
<i>_x</i>	Posición x de la región en la que se realizó la acción.
<i>_y</i>	Posición y de la región en la que se realizó la acción.
<i>_z</i>	Posición z de la región en la que se realizó la acción.
<i>_datetime</i>	Fecha y hora en la que se realizó la acción.

Tabla C.27: Tabla *_actions* de la BD *opensim*.

C.9.2. Tabla *_avatar*

Descripción del contenido de la tabla *_avatar* de la BD *opensim* modificada (Tabla C.28). En esta tabla se almacenan los nombres de los avatares.

CAMPO	DESCRIPCIÓN
<i>_id</i>	UUID del avatar.
<i>first_name</i>	Nombre del avatar.
<i>last_name</i>	Apellido del avatar.

Tabla C.28: Tabla *_avatar* de la BD *opensim*.

C.9.3. Tabla *_bournechannel*

Descripción del contenido de la tabla *_bournechannel* de la BD *opensim* modificada (Tabla C.29). En esta tabla se almacenan registros de las intervenciones en el chat dentro de *OpenSim*. Hay que tener en cuenta que los mensajes al chat pueden ser enviados tanto por avatares como por scripts dentro de Prims (formas básicas que componen todo lo que hay dentro del mundo virtual).

CAMPO	DESCRIPCIÓN
<i>_id</i>	Identificador de la instancia.
<i>_datetime</i>	Fecha y hora en el que se envió el mensaje.
<i>_region</i>	UUID de la región en la que se envió el mensaje.
<i>_channel</i>	Canal del chat por el que se envió el mensaje. <ul style="list-style-type: none"> ■ Canal público - 0. ■ Canales disponibles - (-2147483648, 2147483647).
<i>_chattype</i>	Tipo del mensaje que se envió. El tipo de mensaje depende de la difusión que se quiera dar al mensaje. Los valores que puede tomar son: <ul style="list-style-type: none"> ■ Whisper - De 0 a 10 metros de difusión dentro de la región. ■ Say - De 0 a 20 metros de difusión dentro de la región. ■ Shout - De 0 a 100 metros de difusión dentro de la región. ■ Broadcast - Toda la región.
<i>_x</i>	Posición x de la región en la que se envió el mensaje.
<i>_y</i>	Posición y de la región en la que se envió el mensaje.
<i>_z</i>	Posición z de la región en la que se envió el mensaje.
<i>_speaker</i>	UUID del avatar/prim que envió el mensaje.
<i>_message</i>	Texto del mensaje que se envió.

Tabla C.29: Tabla *_bournechannel* de la BD *opensim*.

C.9.4. Tabla *_bourneim*

Descripción del contenido de la tabla *_bourneim* de la BD *opensim* modificada (Tabla C.30). En esta tabla se almacenan registros de los mensajes instantáneos privados enviados dentro de *OpenSim*. Hay que tener en cuenta que los mensajes pueden ser enviados tanto por avatares como por scripts dentro de Prims (formas básicas que componen todo lo que hay dentro del mundo virtual).

CAMPO	DESCRIPCIÓN
<i>_id</i>	Identificador de la instancia.
<i>_datetime</i>	Fecha y hora en el que se envió el mensaje.
<i>_region</i>	UUID de la región en la que se envió el mensaje.
<i>_speaker</i>	UUID del avatar/prim que envió el mensaje.
<i>_listener</i>	UUID del avatar/prim al que va recibido el mensaje.
<i>_message</i>	Texto del mensaje que se envió.

Tabla C.30: Tabla *_bourneim* de la BD *opensim*.

C.9.5. Tabla *_events*

Descripción del contenido de la tabla *_events* de la BD *opensim* modificada (Tabla C.31). En esta tabla se almacenan registros de los eventos producidos dentro de *OpenSim*.

CAMPO	DESCRIPCIÓN
<i>_id</i>	Identificador de la instancia.
<i>_type</i>	Tipo del evento que se produjo. Los valores que puede tomar son: <ul style="list-style-type: none">■ touch - Tocar un objeto.
<i>_avatar</i>	UUID del avatar que produjo el evento.
<i>_objeto</i>	UUID del objeto con el que se produjo el evento.
<i>_region</i>	UUID de la región en la que se produjo el evento.
<i>_x</i>	Posición x de la región en la que se produjo el evento.
<i>_y</i>	Posición y de la región en la que se produjo el evento.
<i>_z</i>	Posición z de la región en la que se produjo el evento.
<i>_datetime</i>	Fecha y hora en el que se produjo el evento.

Tabla C.31: Tabla *_events* de la BD *opensim*.

C.9.6. Tabla *_regions*

Descripción del contenido de la tabla *_regions* de la BD *opensim* modificada (Tabla C.32). En esta tabla se almacenan los nombres de los regiones.

CAMPO	DESCRIPCIÓN
<i>_id</i>	UUID de la región.
<i>name</i>	Nombre de la región.
<i>_x</i>	Posición x de la región en el mapa de <i>OpenSim</i> .
<i>_y</i>	Posición y de la región en el mapa de <i>OpenSim</i> .

Tabla C.32: Tabla *_regions* de la BD *opensim*.

C.9.7. Tabla *_teleport*

Descripción del contenido de la tabla *_teleport* de la BD *opensim* modificada (Tabla C.33). En esta tabla se almacenan registros de los teletransportes realizados por los avatares en las regiones.

CAMPO	DESCRIPCIÓN
<i>_avatar</i>	UUID del avatar.
<i>_region_id</i>	UUID de la región de origen del teletransporte.
<i>_region_id_dest</i>	UUID de la región de destino del teletransporte.
<i>_x</i>	Posición x de la región a la que se realiza el teletransporte.
<i>_y</i>	Posición y de la región a la que se realiza el teletransporte.
<i>_z</i>	Posición z de la región a la que se realiza el teletransporte.
<i>_datetime</i>	Fecha y hora en el que se produjo el teletransporte.

Tabla C.33: Tabla *_teleport* de la BD *opensim*.

C.9.8. Tabla *_trace*

TABLA EN DESUSO. Descripción del contenido de la tabla *_trace* de la BD *opensim* modificada (Tabla C.34). En esta tabla se registra la posición de un avatar en un determinado instante. Para ello el avatar debe colocarse una pulsera en el mundo virtual, que cada 10 segundos envía la localización exacta del avatar a esta BD.

CAMPO	DESCRIPCIÓN
<i>_avatar</i>	UUID del avatar que envía la información.
<i>_region_id</i>	UUID de la región en la que se encuentra el avatar.
<i>_x</i>	Coordenada X de la región en la que se encuentra el avatar.
<i>_y</i>	Coordenada Y de la región en la que se encuentra el avatar.
<i>_z</i>	Coordenada Z de la región en la que se encuentra el avatar.
<i>_rotation</i>	Rotación del avatar respecto de la región en la que se encuentra.
<i>_datetime</i>	Fecha y hora exactas en las que se almacenó esta instancia en la base de datos.

Tabla C.34: Tabla *_trace* de la BD *opensim*.

Apéndice D

Ejemplo de archivo KTR

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <transformation>
3    <info>
4      <name>ejemplo</name>
5      <description/>
6      <extended_description/>
7      <trans_version/>
8      <trans_type>Normal</trans_type>
9      <trans_status>0</trans_status>
10     <directory>&#x2f;</directory>
11     <parameters></parameters>
12     <log>
13       <trans-log-table>
14         <connection/>
15         <schema/>
16         <table/>
17         <size_limit_lines/>
18         <interval/>
19         <timeout_days/>
20         <field>
21           <id>ID_BATCH</id>
22           <enabled>Y</enabled>
23           <name>ID_BATCH</name>
24         </field>
25         <field>
26           <id>CHANNEL_ID</id>
27           <enabled>Y</enabled>
28           <name>CHANNEL_ID</name>
29         </field>
30         <field>
31           <id>TRANSNAME</id>
32           <enabled>Y</enabled>
33           <name>TRANSNAME</name>
34         </field>
35         <field>
36           <id>STATUS</id>
37           <enabled>Y</enabled>
38           <name>STATUS</name>
39         </field>
40         <field>
41           <id>LINES_READ</id>
42           <enabled>Y</enabled>
43           <name>LINES_READ</name>
```

```

44         <subject/>
45     </field>
46     <field>
47         <id>LINES_WRITTEN</id>
48         <enabled>Y</enabled>
49         <name>LINES_WRITTEN</name>
50         <subject/>
51     </field>
52     <field>
53         <id>LINES_UPDATED</id><enabled>Y</enabled><name>LINES_UPDATED</name><subject/>
54     </field>
55     <field>
56         <id>LINES_INPUT</id><enabled>Y</enabled><name>LINES_INPUT</name><subject/>
57     </field>
58     <field>
59         <id>LINES_OUTPUT</id><enabled>Y</enabled><name>LINES_OUTPUT</name><subject/>
60     </field>
61     <field>
62         <id>LINES_REJECTED</id><enabled>Y</enabled><name>LINES_REJECTED</name><subject/>
63     </field>
64     <field>
65         <id>ERRORS</id><enabled>Y</enabled><name>ERRORS</name>
66     </field>
67     <field>
68         <id>STARTDATE</id><enabled>Y</enabled><name>STARTDATE</name>
69     </field>
70     <field>
71         <id>ENDDATE</id><enabled>Y</enabled><name>ENDDATE</name>
72     </field>
73     <field>
74         <id>LOGDATE</id><enabled>Y</enabled><name>LOGDATE</name>
75     </field>
76     <field>
77         <id>DEPDATE</id><enabled>Y</enabled><name>DEPDATE</name>
78     </field>
79     <field>
80         <id>REPLAYDATE</id><enabled>Y</enabled><name>REPLAYDATE</name>
81     </field>
82     <field>
83         <id>LOG_FIELD</id><enabled>Y</enabled><name>LOG_FIELD</name>
84     </field>
85 </trans-log-table>
86 <perf-log-table>
87     <connection/>
88     <schema/>
89     <table/>
90     <interval/>
91     <timeout_days/>
92     <field><id>ID_BATCH</id><enabled>Y</enabled><name>ID_BATCH</name></field><field><id>
93 </perf-log-table>
94 <channel-log-table>
95     <connection/>
96     <schema/>
97     <table/>
98     <timeout_days/>
99     <field><id>ID_BATCH</id><enabled>Y</enabled><name>ID_BATCH</name></field><field><id>
100 </channel-log-table>
101 <step-log-table>
102     <connection/>
103     <schema/>
104     <table/>

```

```

105         <timeout_days/>
106         <field><id>ID_BATCH</id><enabled>Y</enabled><name>ID_BATCH</name></field><field><id>
107     </step-log-table>
108 </log>
109 <maxdate>
110     <connection/>
111     <table/>
112     <field/>
113     <offset>0.0</offset>
114     <maxdiff>0.0</maxdiff>
115 </maxdate>
116     <size_rowset>10000</size_rowset>
117     <sleep_time_empty>50</sleep_time_empty>
118     <sleep_time_full>50</sleep_time_full>
119     <unique_connections>N</unique_connections>
120     <feedback_shown>Y</feedback_shown>
121     <feedback_size>50000</feedback_size>
122     <using_thread_priorities>Y</using_thread_priorities>
123     <shared_objects_file/>
124     <capture_step_performance>N</capture_step_performance>
125     <step_performance_capturing_delay>1000</step_performance_capturing_delay>
126     <step_performance_capturing_size_limit>100</step_performance_capturing_size_limit>
127     <dependencies></dependencies>
128     <partitionschemas></partitionschemas>
129     <slaveservers></slaveservers>
130     <clusterschemas></clusterschemas>
131     <created_user>-</created_user>
132     <created_date>2012&#47;09&#47;06 12:04:05.323</created_date>
133     <modified_user>-</modified_user>
134     <modified_date>2012&#47;09&#47;06 12:20:41.039</modified_date>
135 </info>
136 <notepads>
137 </notepads>
138 <connection>
139     <name>Memory_HiddenRoom</name>
140     <server>localhost</server>
141     <type>MYSQL</type>
142     <access>Native</access>
143     <database>uca_memory_ordernar_habitacion</database>
144     <port>3306</port>
145     <username>root</username>
146     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
147     <servername/>
148     <data_tablespace/>
149     <index_tablespace/>
150     <attributes>
151         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
152     </attributes>
153 </connection><connection>
154     <name>Memory_Saturn</name>
155     <server>localhost</server>
156     <type>MYSQL</type>
157     <access>Native</access>
158     <database>uca_memory_saturn</database>
159     <port>3306</port>
160     <username>root</username>
161     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
162     <servername/>
163     <data_tablespace/>
164     <index_tablespace/>
165     <attributes>

```

```

166         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
167     </attributes>
168 </connection><connection>
169     <name>HiddenRoom</name>
170     <server>localhost</server>
171     <type>MYSQL</type>
172     <access>Native</access>
173     <database>uca_memory_ordenar_habitacion</database>
174     <port>3306</port>
175     <username>root</username>
176     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
177     <servername/>
178     <data_tablespace/>
179     <index_tablespace/>
180     <attributes>
181         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
182     </attributes>
183 </connection><connection>
184     <name>Saturn</name>
185     <server>localhost</server>
186     <type>MYSQL</type>
187     <access>Native</access>
188     <database>uca_coffeeshop</database>
189     <port>3306</port>
190     <username>root</username>
191     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
192     <servername/>
193     <data_tablespace/>
194     <index_tablespace/>
195     <attributes>
196         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
197     </attributes>
198 </connection><connection>
199     <name>Coffeeshop</name>
200     <server>localhost</server>
201     <type>MYSQL</type>
202     <access>Native</access>
203     <database>uca_cafeteriaversion2</database>
204     <port>3306</port>
205     <username>root</username>
206     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
207     <servername/>
208     <data_tablespace/>
209     <index_tablespace/>
210     <attributes>
211         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
212     </attributes>
213 </connection><connection>
214     <name>Mango</name>
215     <server>localhost</server>
216     <type>MYSQL</type>
217     <access>Native</access>
218     <database>uca_coffeeshop</database>
219     <port>3306</port>
220     <username>root</username>
221     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
222     <servername/>
223     <data_tablespace/>
224     <index_tablespace/>
225     <attributes>
226         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>

```



```

227     </attributes>
228 </connection><connection>
229     <name>os_databases</name>
230     <server>localhost</server>
231     <type>MYSQL</type>
232     <access>Native</access>
233     <database>os_databases</database>
234     <port>3306</port>
235     <username>root</username>
236     <password>Encrypted 2be98afc85b92cbd3fe4aff228da0a8c9</password>
237     <servername/>
238     <data_tablespace/>
239     <index_tablespace/>
240     <attributes>
241         <attribute><code>PORT_NUMBER</code><attribute>3306</attribute></attribute>
242     </attributes>
243 </connection>
244 <order>
245     <hop>
246         <from>TableInput_HiddenRoom_270_250</from>
247         <to>Calculator_HiddenRoom_470_250</to>
248         <enabled>Y</enabled>
249     </hop>
250     <hop>
251         <from>Calculator_HiddenRoom_470_250</from>
252         <to>Sequence_HiddenRoom_670_250</to>
253         <enabled>Y</enabled>
254     </hop>
255     <hop>
256         <from>Sequence_HiddenRoom_670_250</from>
257         <to>SelectValues_HiddenRoom_870_250</to>
258         <enabled>Y</enabled>
259     </hop>
260     <hop>
261         <from>SelectValues_HiddenRoom_870_250</from>
262         <to>JoinRows_HiddenRoom_1070_250</to>
263         <enabled>Y</enabled>
264     </hop>
265     <hop>
266         <from>TableInput_os_databases_1070_150</from>
267         <to>JoinRows_HiddenRoom_1070_250</to>
268         <enabled>Y</enabled>
269     </hop>
270     <hop>
271         <from>JoinRows_HiddenRoom_1070_250</from>
272         <to>SortRows_250</to>
273         <enabled>Y</enabled>
274     </hop>
275     <hop>
276         <from>SortRows_250</from>
277         <to>SelectValues_HiddenRoom_1470_250</to>
278         <enabled>Y</enabled>
279     </hop>
280     <hop>
281         <from>Sequence_HiddenRoom_670_250</from>
282         <to>SelectValues_HiddenRoom_870_350</to>
283         <enabled>Y</enabled>
284     </hop>
285     <hop>
286         <from>SelectValues_HiddenRoom_870_350</from>
287         <to>JoinRows_HiddenRoom_1070_350</to>

```

```

288     <enabled>Y</enabled>
289 </hop>
290 <hop>
291     <from>TableInput_os_databases_1070_150</from>
292     <to>JoinRows_HiddenRoom_1070_350</to>
293     <enabled>Y</enabled>
294 </hop>
295 <hop>
296     <from>JoinRows_HiddenRoom_1070_350</from>
297     <to>SortRows_350</to>
298     <enabled>Y</enabled>
299 </hop>
300 <hop>
301     <from>SortRows_350</from>
302     <to>SelectValues_HiddenRoom_1470_350</to>
303     <enabled>Y</enabled>
304 </hop>
305 <hop>
306     <from>TableInput_Saturn_70_450</from>
307     <to>Calculator_Saturn_70_550</to>
308     <enabled>Y</enabled>
309 </hop>
310 <hop>
311     <from>Calculator_Saturn_70_550</from>
312     <to>FilterRows5_Saturn_270_450</to>
313     <enabled>Y</enabled>
314 </hop>
315 <hop>
316     <from>FilterRows5_Saturn_270_450</from>
317     <to>Dummy3</to>
318     <enabled>Y</enabled>
319 </hop>
320 <hop>
321     <from>Calculator_Saturn_70_550</from>
322     <to>FilterRows6_Saturn_270_550</to>
323     <enabled>Y</enabled>
324 </hop>
325 <hop>
326     <from>FilterRows6_Saturn_270_550</from>
327     <to>Dummy3</to>
328     <enabled>Y</enabled>
329 </hop>
330 <hop>
331     <from>FilterRows5_Saturn_270_450</from>
332     <to>JoinRowsSaturn</to>
333     <enabled>Y</enabled>
334 </hop>
335 <hop>
336     <from>FilterRows6_Saturn_270_550</from>
337     <to>JoinRowsSaturn</to>
338     <enabled>Y</enabled>
339 </hop>
340 <hop>
341     <from>JoinRowsSaturn</from>
342     <to>Calculator_Saturn_470_550</to>
343     <enabled>Y</enabled>
344 </hop>
345 <hop>
346     <from>Calculator_Saturn_470_550</from>
347     <to>FilterRows7_Saturn_670_550</to>
348     <enabled>Y</enabled>

```

```

349     </hop>
350     <hop>
351         <from>FilterRows7_Saturn_670_550</from>
352         <to>Dummy4</to>
353         <enabled>Y</enabled>
354     </hop>
355     <hop>
356         <from>FilterRows7_Saturn_670_550</from>
357         <to>SequenceSaturn</to>
358         <enabled>Y</enabled>
359     </hop>
360     <hop>
361         <from>SequenceSaturn</from>
362         <to>SelectValues_Saturn_870_450</to>
363         <enabled>Y</enabled>
364     </hop>
365     <hop>
366         <from>SelectValues_Saturn_870_450</from>
367         <to>JoinRows_Saturn_1070_450</to>
368         <enabled>Y</enabled>
369     </hop>
370     <hop>
371         <from>TableInput_os_databases_1070_150</from>
372         <to>JoinRows_Saturn_1070_450</to>
373         <enabled>Y</enabled>
374     </hop>
375     <hop>
376         <from>JoinRows_Saturn_1070_450</from>
377         <to>SortRows_450</to>
378         <enabled>Y</enabled>
379     </hop>
380     <hop>
381         <from>SortRows_450</from>
382         <to>SelectValues_Saturn_1470_450</to>
383         <enabled>Y</enabled>
384     </hop>
385     <hop>
386         <from>SequenceSaturn</from>
387         <to>SelectValues_Saturn_870_550</to>
388         <enabled>Y</enabled>
389     </hop>
390     <hop>
391         <from>SelectValues_Saturn_870_550</from>
392         <to>JoinRows_Saturn_1070_550</to>
393         <enabled>Y</enabled>
394     </hop>
395     <hop>
396         <from>TableInput_os_databases_1070_150</from>
397         <to>JoinRows_Saturn_1070_550</to>
398         <enabled>Y</enabled>
399     </hop>
400     <hop>
401         <from>JoinRows_Saturn_1070_550</from>
402         <to>SortRows_550</to>
403         <enabled>Y</enabled>
404     </hop>
405     <hop>
406         <from>SortRows_550</from>
407         <to>SelectValues_Saturn_1470_550</to>
408         <enabled>Y</enabled>
409     </hop>

```

```

410 <hop>
411   <from>TableInput_Coffeeshop_270_650</from>
412   <to>Calculator_Coffeeshop_470_650</to>
413   <enabled>Y</enabled>
414 </hop>
415 <hop>
416   <from>Calculator_Coffeeshop_470_650</from>
417   <to>Sequence_Coffeeshop_670_650</to>
418   <enabled>Y</enabled>
419 </hop>
420 <hop>
421   <from>Sequence_Coffeeshop_670_650</from>
422   <to>SelectValues_Coffeeshop_870_650</to>
423   <enabled>Y</enabled>
424 </hop>
425 <hop>
426   <from>SelectValues_Coffeeshop_870_650</from>
427   <to>JoinRows_Coffeeshop_1070_650</to>
428   <enabled>Y</enabled>
429 </hop>
430 <hop>
431   <from>TableInput_os_databases_1070_150</from>
432   <to>JoinRows_Coffeeshop_1070_650</to>
433   <enabled>Y</enabled>
434 </hop>
435 <hop>
436   <from>JoinRows_Coffeeshop_1070_650</from>
437   <to>SortRows_650</to>
438   <enabled>Y</enabled>
439 </hop>
440 <hop>
441   <from>SortRows_650</from>
442   <to>SelectValues_Coffeeshop_1470_650</to>
443   <enabled>Y</enabled>
444 </hop>
445 <hop>
446   <from>Sequence_Coffeeshop_670_650</from>
447   <to>SelectValues_Coffeeshop_870_750</to>
448   <enabled>Y</enabled>
449 </hop>
450 <hop>
451   <from>SelectValues_Coffeeshop_870_750</from>
452   <to>JoinRows_Coffeeshop_1070_750</to>
453   <enabled>Y</enabled>
454 </hop>
455 <hop>
456   <from>TableInput_os_databases_1070_150</from>
457   <to>JoinRows_Coffeeshop_1070_750</to>
458   <enabled>Y</enabled>
459 </hop>
460 <hop>
461   <from>JoinRows_Coffeeshop_1070_750</from>
462   <to>SortRows_750</to>
463   <enabled>Y</enabled>
464 </hop>
465 <hop>
466   <from>SortRows_750</from>
467   <to>SelectValues_Coffeeshop_1470_750</to>
468   <enabled>Y</enabled>
469 </hop>
470 <hop>

```

```

471     <from>SelectValues_HiddenRoom_1470_250</from>
472     <to>SortedMerge</to>
473     <enabled>Y</enabled>
474 </hop>
475 <hop>
476     <from>SelectValues_HiddenRoom_1470_350</from>
477     <to>SortedMerge</to>
478     <enabled>Y</enabled>
479 </hop>
480 <hop>
481     <from>SelectValues_Saturn_1470_450</from>
482     <to>SortedMerge</to>
483     <enabled>Y</enabled>
484 </hop>
485 <hop>
486     <from>SelectValues_Saturn_1470_550</from>
487     <to>SortedMerge</to>
488     <enabled>Y</enabled>
489 </hop>
490 <hop>
491     <from>SelectValues_Coffeeshop_1470_650</from>
492     <to>SortedMerge</to>
493     <enabled>Y</enabled>
494 </hop>
495 <hop>
496     <from>SelectValues_Coffeeshop_1470_750</from>
497     <to>SortedMerge</to>
498     <enabled>Y</enabled>
499 </hop>
500 <hop>
501     <from>SortedMerge</from>
502     <to>ReplaceString_1870_250</to>
503     <enabled>Y</enabled>
504 </hop>
505 <hop>
506     <from>ReplaceString_1870_250</from>
507     <to>ReplaceString_2070_250</to>
508     <enabled>Y</enabled>
509 </hop>
510 <hop>
511     <from>ReplaceString_2070_250</from>
512     <to>StringOperations_2270_250</to>
513     <enabled>Y</enabled>
514 </hop>
515 <hop>
516     <from>StringOperations_2270_250</from>
517     <to>JoinRowsRegex</to>
518     <enabled>Y</enabled>
519 </hop>
520 <hop>
521     <from>StringOperationsDict</from>
522     <to>JoinRowsRegex</to>
523     <enabled>Y</enabled>
524 </hop>
525 <hop>
526     <from>TextFileInput_2470_450</from>
527     <to>StringOperationsDict</to>
528     <enabled>Y</enabled>
529 </hop>
530 <hop>
531     <from>JoinRowsRegex</from>

```

```

532     <to>UniqueRowsByHashSet_2670_250</to>
533     <enabled>Y</enabled>
534 </hop>
535 <hop>
536     <from>UniqueRowsByHashSet_2670_250</from>
537     <to>ejemplo</to>
538     <enabled>Y</enabled>
539 </hop>
540 </order><step>
541     <name>TableInput_os_databases_1070_150</name>
542     <type>TableInput</type>
543     <description/>
544     <distributed>N</distributed>
545     <custom_distribution/>
546     <copies>1</copies>
547     <partitioning>
548         <method>none</method>
549         <schema_name/>
550     </partitioning>
551     <connection>os_databases</connection>
552     <sql>SELECT _bournechannel._id AS ID&#xa;; CONCAT&#x28;_avatar.first_name
    ,&#x22; &#x22;; _avatar.last_name&#x29; AS NOMBRE&#xa;;
    _bournechannel._message AS MESSAGE&#xa;; _regions._name AS REGION&#xa;;
    _bournechannel._datetime AS DATE&#xa;; FROM _bournechannel, _regions,
    _avatar WHERE _bournechannel._speaker&#x3d;_avatar._id AND
    _bournechannel._region&#x3d;_regions._id
553     &#x26;&#x26; &#x28;_regions._name &#x3d;&#x22;02OrdenarHabitacion&#x22;
    &#x7c;&#x7c; _regions._name&#x3d;&#x22;08CafeteriaVersion2&#x22;
    &#x7c;&#x7c; _regions._name&#x3d;&#x22;04Saturn&#x22; &#x29; &#x26;&#x26;
    &#x28;CONCAT&#x28;_avatar.first_name ,&#x22; &#x22;;
    _avatar.last_name&#x29;&#x3d;&#x22;Alumno 1&#x22; &#x7c;&#x7c;
    CONCAT&#x28;_avatar.first_name ,&#x22; &#x22;;
    _avatar.last_name&#x29;&#x3d;&#x22;Alumno 2&#x22;&#x29;</sql>
554     <limit>0</limit>
555     <lookup/>
556     <execute_each_row>N</execute_each_row>
557     <variables_active>N</variables_active>
558     <lazy_conversion_active>N</lazy_conversion_active>
559     <cluster_schema/>
560     <remotesteps>
561         <input></input>
562         <output></output>
563     </remotesteps>
564     <GUI>
565         <xloc>1070</xloc>
566         <yloc>150</yloc>
567         <draw>Y</draw>
568     </GUI>
569 </step>
570 <step>
571     <name>TableInput_HiddenRoom_270_250</name>
572     <type>TableInput</type>
573     <description/>
574     <distributed>N</distributed>
575     <custom_distribution/>
576     <copies>1</copies>
577     <partitioning>
578         <method>none</method>
579         <schema_name/>
580     </partitioning>
581     <connection>HiddenRoom</connection>

```

```

582 <sql>SELECT mnombre, vnombre, tiempo&#x2a;1000 AS TIME_CALC, tiempo AS TIME,
      timestamp AS FECHA FROM sala_2 WHERE mnombre &#x3d;&#x22;Alumno 1&#x22;
      &#x7c;&#x7c; vnombre &#x3d;&#x22;Alumno 1&#x22; &#x7c;&#x7c; mnombre
      &#x3d;&#x22;Alumno 2&#x22; &#x7c;&#x7c; vnombre &#x3d;&#x22;Alumno
      2&#x22;</sql>
583 <limit>0</limit>
584 <lookup/>
585 <execute_each_row>N</execute_each_row>
586 <variables_active>N</variables_active>
587 <lazy_conversion_active>N</lazy_conversion_active>
588 <cluster_schema/>
589 <remotesteps>
590   <input></input>
591   <output></output>
592 </remotesteps>
593 <GUI>
594   <xloc>270</xloc>
595   <yloc>250</yloc>
596   <draw>Y</draw>
597 </GUI>
598 </step>
599 <step>
600   <name>Calculator_HiddenRoom_470_250</name>
601   <type>Calculator</type>
602   <description/>
603   <distribute>N</distribute>
604   <custom_distribution/>
605   <copies>1</copies>
606   <partitioning>
607     <method>none</method>
608     <schema_name/>
609   </partitioning>
610   <calculation>
611     <field_name>inicio</field_name>
612     <calc_type>SUBTRACT</calc_type>
613     <field_a>FECHA</field_a>
614     <field_b>TIME_CALC</field_b>
615     <field_c/>
616     <value_type>Date</value_type>
617     <value_length>-1</value_length>
618     <value_precision>-1</value_precision>
619     <remove>N</remove>
620     <conversion_mask>yyyy&#x2f;MM&#x2f;dd
        HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
621     <decimal_symbol/>
622     <grouping_symbol/>
623     <currency_symbol/>
624   </calculation>
625   <calculation>
626     <field_name>fin</field_name>
627     <calc_type>COPY_FIELD</calc_type>
628     <field_a>FECHA</field_a>
629     <field_b/>
630     <field_c/>
631     <value_type>Date</value_type>
632     <value_length>-1</value_length>
633     <value_precision>-1</value_precision>
634     <remove>N</remove>
635     <conversion_mask>yyyy&#x2f;MM&#x2f;dd
        HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
636     <decimal_symbol/>

```

```

637         <grouping_symbol/>
638         <currency_symbol/>
639     </calculation>
640     <calculation>
641         <field_name>fin_comp</field_name>
642         <calc_type>COPY_FIELD</calc_type>
643         <field_a>FECHA</field_a>
644         <field_b/>
645         <field_c/>
646         <value_type>Date</value_type>
647         <value_length>-1</value_length>
648         <value_precision>-1</value_precision>
649         <remove>N</remove>
650         <conversion_mask>yyyy&#x2f;MM&#x2f;dd HH&#x3a;mm</conversion_mask>
651         <decimal_symbol/>
652         <grouping_symbol/>
653         <currency_symbol/>
654     </calculation>
655     <cluster_schema/>
656 <remotesteps>
657     <input> </input>
658     <output> </output>
659 </remotesteps>
660 <GUI>
661     <xloc>470</xloc>
662     <yloc>250</yloc>
663     <draw>Y</draw>
664 </GUI>
665 </step>
666 <step>
667     <name>Sequence_HiddenRoom_670_250</name>
668     <type>Sequence</type>
669     <description/>
670     <distributed>N</distributed>
671     <custom_distribution/>
672     <copies>1</copies>
673     <partitioning>
674         <method>none</method>
675         <schema_name/>
676     </partitioning>
677
678     <valuenam>ID_GAME</valuenam>
679     <use_database>N</use_database>
680     <connection/>
681     <schema/>
682     <seqname>SEQ_</seqname>
683     <use_counter>Y</use_counter>
684     <counter_name>SEQ_GAME</counter_name>
685     <start_at>1</start_at>
686     <increment_by>1</increment_by>
687     <max_value>999999999</max_value>
688     <cluster_schema/>
689     <remotesteps> <input> </input> <output> </output> </remotesteps> <GUI>
690         <xloc>670</xloc>
691         <yloc>250</yloc>
692         <draw>Y</draw>
693     </GUI>
694 </step>
695 <step>
696     <name>SelectValues_HiddenRoom_870_250</name>
697     <type>SelectValues</type>

```



```

698     <description/>
699     <distributed>N</distributed>
700     <custom_distribution/>
701     <copies>1</copies>
702     <partitioning>
703         <method>none</method>
704         <schema_name/>
705     </partitioning>
706     <fields>
707         <field>
708             <name>ID_GAME</name>
709         </field>
710         <field>
711             <name>vnombre</name>
712             <rename>NAME</rename>
713         </field>
714         <field>
715             <name>inicio</name>
716         </field>
717         <field>
718             <name>fin</name>
719         </field>
720     <select_unspecified>N</select_unspecified>
721 </fields>
722 <cluster_schema/>
723 <remotesteps>
724     <input></input>
725     <output></output>
726 </remotesteps>
727 <GUI>
728     <xloc>870</xloc>
729     <yloc>250</yloc>
730     <draw>Y</draw>
731 </GUI>
732 </step>
733 <step>
734     <name>JoinRows_HiddenRoom_1070_250</name>
735     <type>JoinRows</type>
736     <description/>
737     <distributed>N</distributed>
738     <custom_distribution/>
739     <copies>1</copies>
740     <partitioning>
741         <method>none</method>
742         <schema_name/>
743     </partitioning>
744     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
745         <prefix>out</prefix>
746         <cache_size>500</cache_size>
747     <main/>
748     <compare>
749     <condition>
750         <negated>N</negated>
751         <conditions>
752         <condition>
753             <negated>N</negated>
754             <leftvalue>NOMBRE</leftvalue>
755             <function>&#x3d;</function>
756             <rightvalue>NAME</rightvalue>
757         </condition>
758     </condition>

```

```

759         <negated>N</negated>
760         <operator>AND</operator>
761         <leftvalue>DATE</leftvalue>
762         <function>&#x3e;&#x3d;</function>
763         <rightvalue>inicio</rightvalue>
764     </condition>
765     <condition>
766         <negated>N</negated>
767         <operator>AND</operator>
768         <leftvalue>DATE</leftvalue>
769         <function>&#x3c;&#x3d;</function>
770         <rightvalue>fin</rightvalue>
771     </condition>
772 </conditions>
773 </condition>
774 </compare>
775     <cluster_schema/>
776     <remotesteps> <input> </input> <output> </output> </remotesteps>
777 <GUI>
778     <xloc>1070</xloc>
779     <yloc>250</yloc>
780     <draw>Y</draw>
781 </GUI>
782 </step>
783 <step>
784     <name>SortRows_250</name>
785     <type>SortRows</type>
786     <description/>
787     <distribute>N</distribute>
788     <custom_distribution/>
789     <copies>1</copies>
790     <partitioning>
791         <method>none</method>
792         <schema_name/>
793     </partitioning>
794     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
795 <prefix>out</prefix>
796 <sort_size>1000000</sort_size>
797 <free_memory/>
798 <compress>N</compress>
799 <compress_variable/>
800 <unique_rows>N</unique_rows>
801 <fields>
802     <field>
803         <name>REGION</name>
804         <ascending>Y</ascending>
805         <case_sensitive>N</case_sensitive>
806         <collator_enabled>N</collator_enabled>
807         <collator_strength>0</collator_strength>
808         <presorted>N</presorted>
809     </field>
810     <field>
811         <name>ID_GAME</name>
812         <ascending>Y</ascending>
813         <case_sensitive>N</case_sensitive>
814         <collator_enabled>N</collator_enabled>
815         <collator_strength>0</collator_strength>
816         <presorted>N</presorted>
817     </field>
818     <field>
819         <name>DATE</name>

```

```

820         <ascending>Y</ascending>
821         <case_sensitive>N</case_sensitive>
822         <collator_enabled>N</collator_enabled>
823         <collator_strength>0</collator_strength>
824         <presorted>N</presorted>
825     </field>
826 </fields>
827 <cluster_schema/>
828 <remotesteps>
829     <input></input>
830     <output></output>
831 </remotesteps>
832 <GUI>
833     <xloc>1270</xloc>
834     <yloc>250</yloc>
835     <draw>Y</draw>
836 </GUI>
837 </step>
838 <step>
839     <name>SelectValues_HiddenRoom_1470_250</name>
840     <type>SelectValues</type>
841     <description/>
842     <distribute>N</distribute>
843     <custom_distribution/>
844     <copies>1</copies>
845     <partitioning>
846         <method>none</method>
847         <schema_name/>
848     </partitioning>
849     <fields>
850         <field>
851             <name>ID_GAME</name>
852         </field>
853         <field>
854             <name>NAME</name>
855         </field>
856         <field>
857             <name>MESSAGE</name>
858         </field>
859         <field>
860             <name>REGION</name>
861         </field>
862         <field>
863             <name>DATE</name>
864         </field>
865     <select_unspecified>N</select_unspecified>
866     </fields>
867     <cluster_schema/>
868     <remotesteps>
869         <input></input>
870         <output></output>
871     </remotesteps>
872     <GUI>
873         <xloc>1470</xloc>
874         <yloc>250</yloc>
875         <draw>Y</draw>
876     </GUI>
877 </step>
878 <step>
879     <name>SelectValues_HiddenRoom_870_350</name>
880     <type>SelectValues</type>

```

```

881     <description/>
882     <distributed>N</distributed>
883     <custom_distribution/>
884     <copies>1</copies>
885     <partitioning>
886         <method>none</method>
887         <schema_name/>
888     </partitioning>
889     <fields>
890         <field>
891             <name>ID_GAME</name>
892         </field>
893         <field>
894             <name>mnombre</name>
895             <rename>NAME</rename>
896         </field>
897         <field>
898             <name>inicio</name>
899         </field>
900         <field>
901             <name>fin</name>
902         </field>
903     <select_unspecified>N</select_unspecified>
904 </fields>
905 <cluster_schema/>
906 <remotesteps>
907     <input></input>
908     <output></output>
909 </remotesteps>
910 <GUI>
911     <xloc>870</xloc>
912     <yloc>350</yloc>
913     <draw>Y</draw>
914 </GUI>
915 </step>
916 <step>
917     <name>JoinRows_HiddenRoom_1070_350</name>
918     <type>JoinRows</type>
919     <description/>
920     <distributed>N</distributed>
921     <custom_distribution/>
922     <copies>1</copies>
923     <partitioning>
924         <method>none</method>
925         <schema_name/>
926     </partitioning>
927     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
928         <prefix>out</prefix>
929         <cache_size>500</cache_size>
930     <main/>
931     <compare>
932     <condition>
933         <negated>N</negated>
934     <conditions>
935     <condition>
936         <negated>N</negated>
937         <leftvalue>NOMBRE</leftvalue>
938         <function>&#x3d;</function>
939         <rightvalue>NAME</rightvalue>
940     </condition>
941     <condition>

```

```

942         <negated>N</negated>
943         <operator>AND</operator>
944         <leftvalue>DATE</leftvalue>
945         <function>&#x3e;&#x3d;</function>
946         <rightvalue>inicio</rightvalue>
947     </condition>
948 </condition>
949     <negated>N</negated>
950     <operator>AND</operator>
951     <leftvalue>DATE</leftvalue>
952     <function>&#x3c;&#x3d;</function>
953     <rightvalue>fin</rightvalue>
954 </condition>
955 </conditions>
956 </condition>
957 </compare>
958 </cluster_schema/>
959 <remotesteps> <input> </input> <output> </output> </remotesteps>
960 <GUI>
961     <xloc>1070</xloc>
962     <yloc>350</yloc>
963     <draw>Y</draw>
964 </GUI>
965 </step>
966 <step>
967     <name>SortRows_350</name>
968     <type>SortRows</type>
969     <description/>
970     <distributed>N</distributed>
971     <custom_distribution/>
972     <copies>1</copies>
973     <partitioning>
974         <method>none</method>
975         <schema_name/>
976     </partitioning>
977     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
978 <prefix>out</prefix>
979 <sort_size>1000000</sort_size>
980 <free_memory/>
981 <compress>N</compress>
982 <compress_variable/>
983 <unique_rows>N</unique_rows>
984 <fields>
985     <field>
986         <name>REGION</name>
987         <ascending>Y</ascending>
988         <case_sensitive>N</case_sensitive>
989         <collator_enabled>N</collator_enabled>
990         <collator_strength>0</collator_strength>
991         <presorted>N</presorted>
992     </field>
993     <field>
994         <name>ID_GAME</name>
995         <ascending>Y</ascending>
996         <case_sensitive>N</case_sensitive>
997         <collator_enabled>N</collator_enabled>
998         <collator_strength>0</collator_strength>
999         <presorted>N</presorted>
1000     </field>
1001     <field>
1002         <name>DATE</name>

```

```

1003         <ascending>Y</ascending>
1004         <case_sensitive>N</case_sensitive>
1005         <collator_enabled>N</collator_enabled>
1006         <collator_strength>0</collator_strength>
1007         <presorted>N</presorted>
1008     </field>
1009 </fields>
1010 <cluster_schema/>
1011 <remotesteps>
1012     <input></input>
1013     <output></output>
1014 </remotesteps>
1015 <GUI>
1016     <xloc>1270</xloc>
1017     <yloc>350</yloc>
1018     <draw>Y</draw>
1019 </GUI>
1020 </step>
1021 <step>
1022     <name>SelectValues_HiddenRoom_1470_350</name>
1023     <type>SelectValues</type>
1024     <description/>
1025     <distributed>N</distributed>
1026     <custom_distribution/>
1027     <copies>1</copies>
1028     <partitioning>
1029         <method>none</method>
1030         <schema_name/>
1031     </partitioning>
1032     <fields>
1033         <field>
1034             <name>ID_GAME</name>
1035         </field>
1036         <field>
1037             <name>NAME</name>
1038         </field>
1039         <field>
1040             <name>MESSAGE</name>
1041         </field>
1042         <field>
1043             <name>REGION</name>
1044         </field>
1045         <field>
1046             <name>DATE</name>
1047         </field>
1048     <select_unspecified>N</select_unspecified>
1049     </fields>
1050 <cluster_schema/>
1051 <remotesteps>
1052     <input></input>
1053     <output></output>
1054 </remotesteps>
1055 <GUI>
1056     <xloc>1470</xloc>
1057     <yloc>350</yloc>
1058     <draw>Y</draw>
1059 </GUI>
1060 </step>
1061 <step>
1062     <name>Dummy3</name>
1063     <type>Dummy</type>

```

```

1064     <description/>
1065     <distribute>N</distribute>
1066     <custom_distribution/>
1067     <copies>1</copies>
1068     <partitioning>
1069         <method>none</method>
1070         <schema_name/>
1071     </partitioning>
1072     <cluster_schema/>
1073     <remotesteps> <input> </input> <output> </output> </remotesteps>
1074 <GUI>
1075     <xloc>270</xloc>
1076     <yloc>150</yloc>
1077     <draw>Y</draw>
1078 </GUI>
1079 </step>
1080 <step>
1081     <name>Dummy4</name>
1082     <type>Dummy</type>
1083     <description/>
1084     <distribute>N</distribute>
1085     <custom_distribution/>
1086     <copies>1</copies>
1087     <partitioning>
1088         <method>none</method>
1089         <schema_name/>
1090     </partitioning>
1091     <cluster_schema/>
1092     <remotesteps> <input> </input> <output> </output> </remotesteps>
1093 <GUI>
1094     <xloc>670</xloc>
1095     <yloc>150</yloc>
1096     <draw>Y</draw>
1097 </GUI>
1098 </step>
1099 <step>
1100     <name>TableInput_Saturn_70_450</name>
1101     <type>TableInput</type>
1102     <description/>
1103     <distribute>N</distribute>
1104     <custom_distribution/>
1105     <copies>1</copies>
1106     <partitioning>
1107         <method>none</method>
1108         <schema_name/>
1109     </partitioning>
1110     <connection>Saturn</connection>
1111     <sql>SELECT nombre,rol, tiempo&#x2a;1000 AS TIME_CALC, tiempo AS TIME,
        timestamp AS FECHA FROM saturn WHERE null&#x3d;&#x22;Alumno 1&#x22;
        &#x7c;&#x7c; null&#x3d;&#x22;Alumno 2&#x22;</sql>
1112     <limit>0</limit>
1113     <lookup/>
1114     <execute_each_row>N</execute_each_row>
1115     <variables_active>N</variables_active>
1116     <lazy_conversion_active>N</lazy_conversion_active>
1117     <cluster_schema/>
1118     <remotesteps>
1119         <input></input>
1120         <output></output>
1121     </remotesteps>
1122 <GUI>

```

```

1123     <xloc>70</xloc>
1124     <yloc>450</yloc>
1125     <draw>Y</draw>
1126   </GUI>
1127 </step>
1128 <step>
1129   <name>Calulator_Saturn_70_550</name>
1130   <type>Calculator</type>
1131   <description/>
1132   <distribute>N</distribute>
1133   <custom_distribution/>
1134   <copies>1</copies>
1135   <partitioning>
1136     <method>none</method>
1137     <schema_name/>
1138   </partitioning>
1139   <calculation>
1140     <field_name>inicio</field_name>
1141     <calc_type>SUBTRACT</calc_type>
1142     <field_a>FECHA</field_a>
1143     <field_b>TIME_CALC</field_b>
1144     <field_c/>
1145     <value_type>Date</value_type>
1146     <value_length>-1</value_length>
1147     <value_precision>-1</value_precision>
1148     <remove>N</remove>
1149     <conversion_mask>yyyy&#x2f;MM&#x2f;dd
1150       HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
1151     <decimal_symbol/>
1152     <grouping_symbol/>
1153     <currency_symbol/>
1154   </calculation>
1155   <calculation>
1156     <field_name>fin</field_name>
1157     <calc_type>COPY_FIELD</calc_type>
1158     <field_a>FECHA</field_a>
1159     <field_b/>
1160     <field_c/>
1161     <value_type>Date</value_type>
1162     <value_length>-1</value_length>
1163     <value_precision>-1</value_precision>
1164     <remove>N</remove>
1165     <conversion_mask>yyyy&#x2f;MM&#x2f;dd
1166       HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
1167     <decimal_symbol/>
1168     <grouping_symbol/>
1169     <currency_symbol/>
1170   </calculation>
1171   <calculation>
1172     <field_name>fin_comp</field_name>
1173     <calc_type>COPY_FIELD</calc_type>
1174     <field_a>FECHA</field_a>
1175     <field_b/>
1176     <field_c/>
1177     <value_type>Date</value_type>
1178     <value_length>-1</value_length>
1179     <value_precision>-1</value_precision>
1180     <remove>N</remove>
1181     <conversion_mask>yyyy&#x2f;MM&#x2f;dd HH&#x3a;mm</conversion_mask>
1182     <decimal_symbol/>
1183     <grouping_symbol/>

```



```

1182         <currency_symbol/>
1183     </calculation>
1184     <cluster_schema/>
1185 <remotesteps>
1186     <input> </input>
1187     <output> </output>
1188 </remotesteps>
1189 <GUI>
1190     <xloc>70</xloc>
1191     <yloc>550</yloc>
1192     <draw>Y</draw>
1193 </GUI>
1194 </step>
1195 <step>
1196     <name>FilterRows5_Saturn_270_450</name>
1197     <type>FilterRows</type>
1198     <description/>
1199     <distribute>N</distribute>
1200     <custom_distribution/>
1201     <copies>1</copies>
1202     <partitioning>
1203         <method>none</method>
1204         <schema_name/>
1205     </partitioning>
1206     <send_true_to>JoinRowsSaturn</send_true_to>
1207 <send_false_to>Dummy3</send_false_to>
1208     <compare>
1209 <condition>
1210     <negated>N</negated>
1211     <conditions>
1212     <condition>
1213         <negated>N</negated>
1214         <leftvalue>rol</leftvalue>
1215         <function>&#x3d;</function>
1216         <rightvalue/>
1217         <value><name>constant</name><type>String</type><text>cliente</text><length>-1</length><precis
1218         </condition>
1219     </conditions>
1220 </condition>
1221     </compare>
1222     <cluster_schema/>
1223 <remotesteps> <input> </input> <output> </output> </remotesteps>
1224 <GUI>
1225     <xloc>270</xloc>
1226     <yloc>450</yloc>
1227     <draw>Y</draw>
1228 </GUI>
1229 </step>
1230 <step>
1231     <name>FilterRows6_Saturn_270_550</name>
1232     <type>FilterRows</type>
1233     <description/>
1234     <distribute>N</distribute>
1235     <custom_distribution/>
1236     <copies>1</copies>
1237     <partitioning>
1238         <method>none</method>
1239         <schema_name/>
1240     </partitioning>
1241     <send_true_to>JoinRowsSaturn</send_true_to>
1242 <send_false_to>Dummy3</send_false_to>

```

```

1242     <compare>
1243 <condition>
1244   <negated>N</negated>
1245   <conditions>
1246     <condition>
1247       <negated>N</negated>
1248       <leftvalue>rol</leftvalue>
1249       <function>&#x3d;</function>
1250       <rightvalue/>
1251       <value><name>constant</name><type>String</type><text>vendedor</text><length>-1</length><preci
1252     </condition>
1253   </conditions>
1254 </condition>
1255 </compare>
1256 <cluster_schema/>
1257 <remotesteps> <input> </input> <output> </output> </remotesteps>
1258 <GUI>
1259   <xloc>270</xloc>
1260   <yloc>550</yloc>
1261   <draw>Y</draw>
1262 </GUI>
1263 </step>
1264 <step>
1265   <name>JoinRowsSaturn</name>
1266   <type>JoinRows</type>
1267   <description/>
1268   <distributed>N</distributed>
1269   <custom_distribution/>
1270   <copies>1</copies>
1271   <partitioning>
1272     <method>none</method>
1273     <schema_name/>
1274   </partitioning>
1275   <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
1276     <prefix>out</prefix>
1277     <cache_size>500000</cache_size>
1278     <main>FilterRows5_Saturn_270_450</main>
1279     <compare>
1280       <condition>
1281         <negated>N</negated>
1282         <conditions>
1283           <condition>
1284             <negated>N</negated>
1285             <operator>AND</operator>
1286             <leftvalue>fin</leftvalue>
1287             <function>IS NOT NULL</function>
1288             <rightvalue></rightvalue>
1289           </condition>
1290         </conditions>
1291       </condition>
1292     </compare>
1293     <cluster_schema/>
1294     <remotesteps> <input> </input> <output> </output> </remotesteps>
1295   <GUI>
1296     <xloc>470</xloc>
1297     <yloc>450</yloc>
1298     <draw>Y</draw>
1299   </GUI>
1300 </step>
1301 <step>
1302   <name>Calulator_Saturn_470_550</name>

```

```

1302     <type>Calculator</type>
1303     <description/>
1304     <distributed>N</distributed>
1305     <custom_distribution/>
1306     <copies>1</copies>
1307     <partitioning>
1308         <method>none</method>
1309         <schema_name/>
1310     </partitioning>
1311     <calculation>
1312         <field_name>comp</field_name>
1313         <calc_type>DATE_DIFF_MN</calc_type>
1314         <field_a>fin</field_a>
1315         <field_b>fin_1</field_b>
1316         <field_c/>
1317         <value_type>Integer</value_type>
1318         <value_length>-1</value_length>
1319         <value_precision>-1</value_precision>
1320         <remove>N</remove>
1321         <conversion_mask>&#x23;</conversion_mask>
1322         <decimal_symbol/>
1323         <grouping_symbol/>
1324         <currency_symbol/>
1325     </calculation><cluster_schema/>
1326 <remotesteps>
1327     <input> </input>
1328     <output> </output>
1329 </remotesteps>
1330 <GUI>
1331     <xloc>470</xloc>
1332     <yloc>550</yloc>
1333     <draw>Y</draw>
1334 </GUI>
1335 </step>
1336 <step>
1337     <name>FilterRows7_Saturn_670_550</name>
1338     <type>FilterRows</type>
1339     <description/>
1340     <distributed>N</distributed>
1341     <custom_distribution/>
1342     <copies>1</copies>
1343     <partitioning>
1344         <method>none</method>
1345         <schema_name/>
1346     </partitioning>
1347     <send_true_to>SequenceSaturn</send_true_to>
1348 <send_false_to>Dummy4</send_false_to>
1349     <compare>
1350 <condition>
1351     <negated>N</negated>
1352     <conditions>
1353     <condition>
1354     <negated>N</negated>
1355     <leftvalue>comp</leftvalue>
1356     <function>&#x3d;</function>
1357     <rightvalue/>
1358     <value><name>constant</name><type>Integer</type><text>0</text><length>-1</length><precision>0
1359     </condition>
1360     </conditions>
1361 </condition>
1362 </compare>

```

```

1362     <cluster_schema/>
1363     <remotesteps> <input> </input> <output> </output> </remotesteps>
1364 <GUI>
1365     <xloc>670</xloc>
1366     <yloc>550</yloc>
1367     <draw>Y</draw>
1368 </GUI>
1369 </step>
1370 <step>
1371     <name>SequenceSaturn</name>
1372     <type>Sequence</type>
1373     <description/>
1374     <distribute>N</distribute>
1375     <custom_distribution/>
1376     <copies>1</copies>
1377     <partitioning>
1378         <method>none</method>
1379         <schema_name/>
1380     </partitioning>
1381
1382         <valuenam>ID_GAME</valuenam>
1383         <use_database>N</use_database>
1384         <connection/>
1385         <schema/>
1386         <seqname>SEQ_</seqname>
1387         <use_counter>Y</use_counter>
1388         <counter_name>SEQ_GAME</counter_name>
1389         <start_at>1</start_at>
1390         <increment_by>1</increment_by>
1391         <max_value>999999999</max_value>
1392     <cluster_schema/>
1393     <remotesteps> <input> </input> <output> </output> </remotesteps> <GUI>
1394         <xloc>670</xloc>
1395         <yloc>450</yloc>
1396         <draw>Y</draw>
1397     </GUI>
1398 </step>
1399 <step>
1400     <name>SelectValues_Saturn_870_450</name>
1401     <type>SelectValues</type>
1402     <description/>
1403     <distribute>N</distribute>
1404     <custom_distribution/>
1405     <copies>1</copies>
1406     <partitioning>
1407         <method>none</method>
1408         <schema_name/>
1409     </partitioning>
1410     <fields>
1411         <field>
1412             <name>ID_GAME</name>
1413         </field>
1414         <field>
1415             <name>nombre</name>
1416             <rename>NAME</rename>
1417         </field>
1418         <field>
1419             <name>inicio</name>
1420         </field>
1421         <field>
1422             <name>fin</name>

```

```

1423         </field>
1424     <select_unspecified>N</select_unspecified>
1425 </fields>
1426 <cluster_schema/>
1427 <remotesteps>
1428     <input></input>
1429     <output></output>
1430 </remotesteps>
1431 <GUI>
1432     <xloc>870</xloc>
1433     <yloc>450</yloc>
1434     <draw>Y</draw>
1435 </GUI>
1436 </step>
1437 <step>
1438     <name>JoinRows_Saturn_1070_450</name>
1439     <type>JoinRows</type>
1440     <description/>
1441     <distribute>N</distribute>
1442     <custom_distribution/>
1443     <copies>1</copies>
1444     <partitioning>
1445         <method>none</method>
1446         <schema_name/>
1447     </partitioning>
1448     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
1449         <prefix>out</prefix>
1450         <cache_size>500</cache_size>
1451         <main/>
1452         <compare>
1453             <condition>
1454                 <negated>N</negated>
1455                 <conditions>
1456                     <condition>
1457                         <negated>N</negated>
1458                         <leftvalue>NOMBRE</leftvalue>
1459                         <function>&#x3d;</function>
1460                         <rightvalue>NAME</rightvalue>
1461                     </condition>
1462                     <condition>
1463                         <negated>N</negated>
1464                         <operator>AND</operator>
1465                         <leftvalue>DATE</leftvalue>
1466                         <function>&#x3e;&#x3d;</function>
1467                         <rightvalue>inicio</rightvalue>
1468                     </condition>
1469                     <condition>
1470                         <negated>N</negated>
1471                         <operator>AND</operator>
1472                         <leftvalue>DATE</leftvalue>
1473                         <function>&#x3c;&#x3d;</function>
1474                         <rightvalue>fin</rightvalue>
1475                     </condition>
1476                 </conditions>
1477             </condition>
1478         </compare>
1479         <cluster_schema/>
1480     <remotesteps> <input> </input> <output> </output> </remotesteps>
1481 <GUI>
1482     <xloc>1070</xloc>
1483     <yloc>450</yloc>

```

```

1484         <draw>Y</draw>
1485     </GUI>
1486 </step>
1487 <step>
1488     <name>SortRows_450</name>
1489     <type>SortRows</type>
1490     <description/>
1491     <distributed>N</distributed>
1492     <custom_distribution/>
1493     <copies>1</copies>
1494     <partitioning>
1495         <method>none</method>
1496         <schema_name/>
1497     </partitioning>
1498     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
1499     <prefix>out</prefix>
1500     <sort_size>1000000</sort_size>
1501     <free_memory/>
1502     <compress>N</compress>
1503     <compress_variable/>
1504     <unique_rows>N</unique_rows>
1505     <fields>
1506         <field>
1507             <name>REGION</name>
1508             <ascending>Y</ascending>
1509             <case_sensitive>N</case_sensitive>
1510             <collator_enabled>N</collator_enabled>
1511             <collator_strength>0</collator_strength>
1512             <presorted>N</presorted>
1513         </field>
1514         <field>
1515             <name>ID_GAME</name>
1516             <ascending>Y</ascending>
1517             <case_sensitive>N</case_sensitive>
1518             <collator_enabled>N</collator_enabled>
1519             <collator_strength>0</collator_strength>
1520             <presorted>N</presorted>
1521         </field>
1522         <field>
1523             <name>DATE</name>
1524             <ascending>Y</ascending>
1525             <case_sensitive>N</case_sensitive>
1526             <collator_enabled>N</collator_enabled>
1527             <collator_strength>0</collator_strength>
1528             <presorted>N</presorted>
1529         </field>
1530     </fields>
1531     <cluster_schema/>
1532     <remotesteps>
1533         <input></input>
1534         <output></output>
1535     </remotesteps>
1536     <GUI>
1537         <xloc>1270</xloc>
1538         <yloc>450</yloc>
1539         <draw>Y</draw>
1540     </GUI>
1541 </step>
1542 <step>
1543     <name>SelectValues_Saturn_1470_450</name>
1544     <type>SelectValues</type>

```

```

1545     <description/>
1546     <distributed>N</distributed>
1547     <custom_distribution/>
1548     <copies>1</copies>
1549     <partitioning>
1550         <method>none</method>
1551         <schema_name/>
1552     </partitioning>
1553     <fields>
1554         <field>
1555             <name>ID_GAME</name>
1556         </field>
1557         <field>
1558             <name>NAME</name>
1559         </field>
1560         <field>
1561             <name>MESSAGE</name>
1562         </field>
1563         <field>
1564             <name>REGION</name>
1565         </field>
1566         <field>
1567             <name>DATE</name>
1568         </field>
1569     <select_unspecified>N</select_unspecified>
1570 </fields>
1571 <cluster_schema/>
1572 <remotesteps>
1573     <input></input>
1574     <output></output>
1575 </remotesteps>
1576 <GUI>
1577     <xloc>1470</xloc>
1578     <yloc>450</yloc>
1579     <draw>Y</draw>
1580 </GUI>
1581 </step>
1582 <step>
1583     <name>SelectValues_Saturn_870_550</name>
1584     <type>SelectValues</type>
1585     <description/>
1586     <distributed>N</distributed>
1587     <custom_distribution/>
1588     <copies>1</copies>
1589     <partitioning>
1590         <method>none</method>
1591         <schema_name/>
1592     </partitioning>
1593     <fields>
1594         <field>
1595             <name>ID_GAME</name>
1596         </field>
1597         <field>
1598             <name>nombre_1</name>
1599             <rename>NAME</rename>
1600         </field>
1601         <field>
1602             <name>inicio</name>
1603         </field>
1604         <field>
1605             <name>fin</name>

```

```

1606         </field>
1607     <select_unspecified>N</select_unspecified>
1608 </fields>
1609 <cluster_schema/>
1610 <remotesteps>
1611     <input></input>
1612     <output></output>
1613 </remotesteps>
1614 <GUI>
1615     <xloc>870</xloc>
1616     <yloc>550</yloc>
1617     <draw>Y</draw>
1618 </GUI>
1619 </step>
1620 <step>
1621     <name>JoinRows_Saturn_1070_550</name>
1622     <type>JoinRows</type>
1623     <description/>
1624     <distributed>N</distributed>
1625     <custom_distribution/>
1626     <copies>1</copies>
1627     <partitioning>
1628         <method>none</method>
1629         <schema_name/>
1630     </partitioning>
1631     <directory>#{x25};#{x25}; java.io.tmpdir#{x25};#{x25};</directory>
1632         <prefix>out</prefix>
1633         <cache_size>500</cache_size>
1634         <main/>
1635         <compare>
1636             <condition>
1637                 <negated>N</negated>
1638                 <conditions>
1639                     <condition>
1640                         <negated>N</negated>
1641                         <leftvalue>NOMBRE</leftvalue>
1642                         <function>#{x3d}</function>
1643                         <rightvalue>NAME</rightvalue>
1644                     </condition>
1645                     <condition>
1646                         <negated>N</negated>
1647                         <operator>AND</operator>
1648                         <leftvalue>DATE</leftvalue>
1649                         <function>#{x3e};#{x3d}</function>
1650                         <rightvalue>inicio</rightvalue>
1651                     </condition>
1652                     <condition>
1653                         <negated>N</negated>
1654                         <operator>AND</operator>
1655                         <leftvalue>DATE</leftvalue>
1656                         <function>#{x3c};#{x3d}</function>
1657                         <rightvalue>fin</rightvalue>
1658                     </condition>
1659                 </conditions>
1660             </condition>
1661         </compare>
1662         <cluster_schema/>
1663     <remotesteps> <input> </input> <output> </output> </remotesteps>
1664 <GUI>
1665     <xloc>1070</xloc>
1666     <yloc>550</yloc>

```



```

1667         <draw>Y</draw>
1668     </GUI>
1669 </step>
1670 <step>
1671     <name>SortRows_550</name>
1672     <type>SortRows</type>
1673     <description/>
1674     <distributed>N</distributed>
1675     <custom_distribution/>
1676     <copies>1</copies>
1677     <partitioning>
1678         <method>none</method>
1679         <schema_name/>
1680     </partitioning>
1681     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
1682 <prefix>out</prefix>
1683 <sort_size>1000000</sort_size>
1684 <free_memory/>
1685 <compress>N</compress>
1686 <compress_variable/>
1687 <unique_rows>N</unique_rows>
1688 <fields>
1689     <field>
1690         <name>REGION</name>
1691         <ascending>Y</ascending>
1692         <case_sensitive>N</case_sensitive>
1693         <collator_enabled>N</collator_enabled>
1694         <collator_strength>0</collator_strength>
1695         <presorted>N</presorted>
1696     </field>
1697     <field>
1698         <name>ID_GAME</name>
1699         <ascending>Y</ascending>
1700         <case_sensitive>N</case_sensitive>
1701         <collator_enabled>N</collator_enabled>
1702         <collator_strength>0</collator_strength>
1703         <presorted>N</presorted>
1704     </field>
1705     <field>
1706         <name>DATE</name>
1707         <ascending>Y</ascending>
1708         <case_sensitive>N</case_sensitive>
1709         <collator_enabled>N</collator_enabled>
1710         <collator_strength>0</collator_strength>
1711         <presorted>N</presorted>
1712     </field>
1713 </fields>
1714 <cluster_schema/>
1715 <remotesteps>
1716     <input></input>
1717     <output></output>
1718 </remotesteps>
1719 <GUI>
1720     <xloc>1270</xloc>
1721     <yloc>550</yloc>
1722     <draw>Y</draw>
1723 </GUI>
1724 </step>
1725 <step>
1726     <name>SelectValues_Saturn_1470_550</name>
1727     <type>SelectValues</type>

```

```

1728     <description/>
1729     <distributed>N</distributed>
1730     <custom_distribution/>
1731     <copies>1</copies>
1732     <partitioning>
1733         <method>none</method>
1734         <schema_name/>
1735     </partitioning>
1736     <fields>
1737         <field>
1738             <name>ID_GAME</name>
1739         </field>
1740         <field>
1741             <name>NAME</name>
1742         </field>
1743         <field>
1744             <name>MESSAGE</name>
1745         </field>
1746         <field>
1747             <name>REGION</name>
1748         </field>
1749         <field>
1750             <name>DATE</name>
1751         </field>
1752     <select_unspecified>N</select_unspecified>
1753     </fields>
1754     <cluster_schema/>
1755     <remotesteps>
1756         <input></input>
1757         <output></output>
1758     </remotesteps>
1759     <GUI>
1760         <xloc>1470</xloc>
1761         <yloc>550</yloc>
1762         <draw>Y</draw>
1763     </GUI>
1764 </step>
1765 <step>
1766     <name>TableInput_Coffeeshop_270_650</name>
1767     <type>TableInput</type>
1768     <description/>
1769     <distributed>N</distributed>
1770     <custom_distribution/>
1771     <copies>1</copies>
1772     <partitioning>
1773         <method>none</method>
1774         <schema_name/>
1775     </partitioning>
1776     <connection>Coffeeshop</connection>
1777     <sql>SELECT nombre_cam1, nombre_cam2, tiempo&#x2a;1000 AS TIME_CALC, tiempo AS
        TIME, timestamp AS FECHA FROM partida WHERE nombre_cam1 &#x3d;&#x22;Alumno
        1&#x22;; &#x7c;&#x7c; nombre_cam2 &#x3d;&#x22;Alumno 1&#x22;; &#x7c;&#x7c;
        nombre_cam1 &#x3d;&#x22;Alumno 2&#x22;; &#x7c;&#x7c; nombre_cam2
        &#x3d;&#x22;Alumno 2&#x22;;</sql>
1778     <limit>0</limit>
1779     <lookup/>
1780     <execute_each_row>N</execute_each_row>
1781     <variables_active>N</variables_active>
1782     <lazy_conversion_active>N</lazy_conversion_active>
1783     <cluster_schema/>
1784     <remotesteps>

```

```

1785     <input></input>
1786     <output></output>
1787 </remotesteps>
1788 <GUI>
1789     <xloc>270</xloc>
1790     <yloc>650</yloc>
1791     <draw>Y</draw>
1792 </GUI>
1793 </step>
1794 <step>
1795     <name>Calculator_Coffeeshop_470_650</name>
1796     <type>Calculator</type>
1797     <description/>
1798     <distribute>N</distribute>
1799     <custom_distribution/>
1800     <copies>1</copies>
1801     <partitioning>
1802         <method>none</method>
1803         <schema_name/>
1804     </partitioning>
1805     <calculation>
1806         <field_name>inicio</field_name>
1807         <calc_type>SUBTRACT</calc_type>
1808         <field_a>FECHA</field_a>
1809         <field_b>TIME_CALC</field_b>
1810         <field_c/>
1811         <value_type>Date</value_type>
1812         <value_length>-1</value_length>
1813         <value_precision>-1</value_precision>
1814         <remove>N</remove>
1815         <conversion_mask>yyyy&#x2f;MM&#x2f;dd
1816             HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
1817         <decimal_symbol/>
1818         <grouping_symbol/>
1819         <currency_symbol/>
1820     </calculation>
1821     <calculation>
1822         <field_name>fin</field_name>
1823         <calc_type>COPY_FIELD</calc_type>
1824         <field_a>FECHA</field_a>
1825         <field_b/>
1826         <field_c/>
1827         <value_type>Date</value_type>
1828         <value_length>-1</value_length>
1829         <value_precision>-1</value_precision>
1830         <remove>N</remove>
1831         <conversion_mask>yyyy&#x2f;MM&#x2f;dd
1832             HH&#x3a;mm&#x3a;ss.SSS</conversion_mask>
1833         <decimal_symbol/>
1834         <grouping_symbol/>
1835         <currency_symbol/>
1836     </calculation>
1837     <calculation>
1838         <field_name>fin_comp</field_name>
1839         <calc_type>COPY_FIELD</calc_type>
1840         <field_a>FECHA</field_a>
1841         <field_b/>
1842         <field_c/>
1843         <value_type>Date</value_type>
1844         <value_length>-1</value_length>
1845         <value_precision>-1</value_precision>

```

```

1844         <remove>N</remove>
1845         <conversion_mask>yyyy&#x2f;MM&#x2f;dd HH&#x3a;mm</conversion_mask>
1846         <decimal_symbol/>
1847         <grouping_symbol/>
1848         <currency_symbol/>
1849     </calculation>
1850     <cluster_schema/>
1851 <remotesteps>
1852     <input> </input>
1853     <output> </output>
1854 </remotesteps>
1855 <GUI>
1856     <xloc>470</xloc>
1857     <yloc>650</yloc>
1858     <draw>Y</draw>
1859 </GUI>
1860 </step>
1861 <step>
1862     <name>Sequence_Coffeeshop_670_650</name>
1863     <type>Sequence</type>
1864     <description/>
1865     <distributed>N</distributed>
1866     <custom_distribution/>
1867     <copies>1</copies>
1868     <partitioning>
1869         <method>none</method>
1870         <schema_name/>
1871     </partitioning>
1872
1873     <valuenam>ID_GAME</valuenam>
1874     <use_database>N</use_database>
1875     <connection/>
1876     <schema/>
1877     <seqname>SEQ_</seqname>
1878     <use_counter>Y</use_counter>
1879     <counter_name>SEQ_GAME</counter_name>
1880     <start_at>1</start_at>
1881     <increment_by>1</increment_by>
1882     <max_value>999999999</max_value>
1883     <cluster_schema/>
1884     <remotesteps> <input> </input> <output> </output> </remotesteps> <GUI>
1885         <xloc>670</xloc>
1886         <yloc>650</yloc>
1887         <draw>Y</draw>
1888     </GUI>
1889 </step>
1890 <step>
1891     <name>SelectValues_Coffeeshop_870_650</name>
1892     <type>SelectValues</type>
1893     <description/>
1894     <distributed>N</distributed>
1895     <custom_distribution/>
1896     <copies>1</copies>
1897     <partitioning>
1898         <method>none</method>
1899         <schema_name/>
1900     </partitioning>
1901     <fields>
1902         <field>
1903             <name>ID_GAME</name>
1904         </field>

```

```

1905         <field>
1906             <name>nombre_caml</name>
1907             <rename>NAME</rename>
1908         </field>
1909         <field>
1910             <name>inicio</name>
1911         </field>
1912         <field>
1913             <name>fin</name>
1914         </field>
1915     <select_unspecified>N</select_unspecified>
1916 </fields>
1917 <cluster_schema/>
1918 <remotesteps>
1919     <input></input>
1920     <output></output>
1921 </remotesteps>
1922 <GUI>
1923     <xloc>870</xloc>
1924     <yloc>650</yloc>
1925     <draw>Y</draw>
1926 </GUI>
1927 </step>
1928 <step>
1929     <name>JoinRows_Coffeeshop_1070_650</name>
1930     <type>JoinRows</type>
1931     <description/>
1932     <distributed>N</distributed>
1933     <custom_distribution/>
1934     <copies>1</copies>
1935     <partitioning>
1936         <method>none</method>
1937         <schema_name/>
1938     </partitioning>
1939     <directory>&#x25;&#x25; java.io.tmpdir&#x25;&#x25;</directory>
1940         <prefix>out</prefix>
1941         <cache_size>500</cache_size>
1942         <main/>
1943         <compare>
1944             <condition>
1945                 <negated>N</negated>
1946                 <conditions>
1947                     <condition>
1948                         <negated>N</negated>
1949                         <leftvalue>NOMBRE</leftvalue>
1950                         <function>&#x3d;</function>
1951                         <rightvalue>NAME</rightvalue>
1952                     </condition>
1953                     <condition>
1954                         <negated>N</negated>
1955                         <operator>AND</operator>
1956                         <leftvalue>DATE</leftvalue>
1957                         <function>&#x3e;&#x3d;</function>
1958                         <rightvalue>inicio</rightvalue>
1959                     </condition>
1960                     <condition>
1961                         <negated>N</negated>
1962                         <operator>AND</operator>
1963                         <leftvalue>DATE</leftvalue>
1964                         <function>&#x3c;&#x3d;</function>
1965                         <rightvalue>fin</rightvalue>

```

```

1966         </condition>
1967     </conditions>
1968 </condition>
1969 </compare>
1970     <cluster_schema/>
1971 <remotesteps> <input> </input> <output> </output> </remotesteps>
1972 <GUI>
1973     <xloc>1070</xloc>
1974     <yloc>650</yloc>
1975     <draw>Y</draw>
1976 </GUI>
1977 </step>
1978 <step>
1979     <name>SortRows_650</name>
1980     <type>SortRows</type>
1981     <description/>
1982     <distribute>N</distribute>
1983     <custom_distribution/>
1984     <copies>1</copies>
1985     <partitioning>
1986         <method>none</method>
1987         <schema_name/>
1988     </partitioning>
1989     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
1990 <prefix>out</prefix>
1991 <sort_size>1000000</sort_size>
1992 <free_memory/>
1993 <compress>N</compress>
1994 <compress_variable/>
1995 <unique_rows>N</unique_rows>
1996 <fields>
1997     <field>
1998         <name>REGION</name>
1999         <ascending>Y</ascending>
2000         <case_sensitive>N</case_sensitive>
2001         <collator_enabled>N</collator_enabled>
2002         <collator_strength>0</collator_strength>
2003         <presorted>N</presorted>
2004     </field>
2005     <field>
2006         <name>ID_GAME</name>
2007         <ascending>Y</ascending>
2008         <case_sensitive>N</case_sensitive>
2009         <collator_enabled>N</collator_enabled>
2010         <collator_strength>0</collator_strength>
2011         <presorted>N</presorted>
2012     </field>
2013     <field>
2014         <name>DATE</name>
2015         <ascending>Y</ascending>
2016         <case_sensitive>N</case_sensitive>
2017         <collator_enabled>N</collator_enabled>
2018         <collator_strength>0</collator_strength>
2019         <presorted>N</presorted>
2020     </field>
2021 </fields>
2022 <cluster_schema/>
2023 <remotesteps>
2024     <input></input>
2025     <output></output>
2026 </remotesteps>

```

```

2027 <GUI>
2028   <xloc>1270</xloc>
2029   <yloc>650</yloc>
2030   <draw>Y</draw>
2031 </GUI>
2032 </step>
2033 <step>
2034   <name>SelectValues_Coffeeshop_1470_650</name>
2035   <type>SelectValues</type>
2036   <description/>
2037   <distributed>N</distributed>
2038   <custom_distribution/>
2039   <copies>1</copies>
2040   <partitioning>
2041     <method>none</method>
2042     <schema_name/>
2043   </partitioning>
2044   <fields>
2045     <field>
2046       <name>ID_GAME</name>
2047     </field>
2048     <field>
2049       <name>NAME</name>
2050     </field>
2051     <field>
2052       <name>MESSAGE</name>
2053     </field>
2054     <field>
2055       <name>REGION</name>
2056     </field>
2057     <field>
2058       <name>DATE</name>
2059     </field>
2060   <select_unspecified>N</select_unspecified>
2061   </fields>
2062   <cluster_schema/>
2063   <remotesteps>
2064     <input></input>
2065     <output></output>
2066   </remotesteps>
2067   <GUI>
2068     <xloc>1470</xloc>
2069     <yloc>650</yloc>
2070     <draw>Y</draw>
2071   </GUI>
2072 </step>
2073 <step>
2074   <name>SelectValues_Coffeeshop_870_750</name>
2075   <type>SelectValues</type>
2076   <description/>
2077   <distributed>N</distributed>
2078   <custom_distribution/>
2079   <copies>1</copies>
2080   <partitioning>
2081     <method>none</method>
2082     <schema_name/>
2083   </partitioning>
2084   <fields>
2085     <field>
2086       <name>ID_GAME</name>
2087     </field>

```

```

2088         <field>
2089             <name>nombre_cam2</name>
2090             <rename>NAME</rename>
2091         </field>
2092         <field>
2093             <name>inicio</name>
2094         </field>
2095         <field>
2096             <name>fin</name>
2097         </field>
2098     <select_unspecified>N</select_unspecified>
2099 </fields>
2100 <cluster_schema/>
2101 <remotesteps>
2102     <input></input>
2103     <output></output>
2104 </remotesteps>
2105 <GUI>
2106     <xloc>870</xloc>
2107     <yloc>750</yloc>
2108     <draw>Y</draw>
2109 </GUI>
2110 </step>
2111 <step>
2112     <name>JoinRows_Coffeeshop_1070_750</name>
2113     <type>JoinRows</type>
2114     <description/>
2115     <distributed>N</distributed>
2116     <custom_distribution/>
2117     <copies>1</copies>
2118     <partitioning>
2119         <method>none</method>
2120         <schema_name/>
2121     </partitioning>
2122     <directory>&#x25;&#x25; java.io.tmpdir&#x25;&#x25;</directory>
2123         <prefix>out</prefix>
2124         <cache_size>500</cache_size>
2125         <main/>
2126     <compare>
2127     <condition>
2128         <negated>N</negated>
2129         <conditions>
2130             <condition>
2131                 <negated>N</negated>
2132                 <leftvalue>NOMBRE</leftvalue>
2133                 <function>&#x3d;</function>
2134                 <rightvalue>NAME</rightvalue>
2135             </condition>
2136             <condition>
2137                 <negated>N</negated>
2138                 <operator>AND</operator>
2139                 <leftvalue>DATE</leftvalue>
2140                 <function>&#x3e;&#x3d;</function>
2141                 <rightvalue>inicio</rightvalue>
2142             </condition>
2143             <condition>
2144                 <negated>N</negated>
2145                 <operator>AND</operator>
2146                 <leftvalue>DATE</leftvalue>
2147                 <function>&#x3c;&#x3d;</function>
2148                 <rightvalue>fin</rightvalue>

```



```

2149         </condition>
2150     </conditions>
2151 </condition>
2152 </compare>
2153     <cluster_schema/>
2154     <remotesteps> <input> </input> <output> </output> </remotesteps>
2155 <GUI>
2156     <xloc>1070</xloc>
2157     <yloc>750</yloc>
2158     <draw>Y</draw>
2159 </GUI>
2160 </step>
2161 <step>
2162     <name>SortRows_750</name>
2163     <type>SortRows</type>
2164     <description/>
2165     <distributed>N</distributed>
2166     <custom_distribution/>
2167     <copies>1</copies>
2168     <partitioning>
2169         <method>none</method>
2170         <schema_name/>
2171     </partitioning>
2172     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
2173 <prefix>out</prefix>
2174 <sort_size>1000000</sort_size>
2175 <free_memory/>
2176 <compress>N</compress>
2177 <compress_variable/>
2178 <unique_rows>N</unique_rows>
2179 <fields>
2180     <field>
2181         <name>REGION</name>
2182         <ascending>Y</ascending>
2183         <case_sensitive>N</case_sensitive>
2184         <collator_enabled>N</collator_enabled>
2185         <collator_strength>0</collator_strength>
2186         <presorted>N</presorted>
2187     </field>
2188     <field>
2189         <name>ID_GAME</name>
2190         <ascending>Y</ascending>
2191         <case_sensitive>N</case_sensitive>
2192         <collator_enabled>N</collator_enabled>
2193         <collator_strength>0</collator_strength>
2194         <presorted>N</presorted>
2195     </field>
2196     <field>
2197         <name>DATE</name>
2198         <ascending>Y</ascending>
2199         <case_sensitive>N</case_sensitive>
2200         <collator_enabled>N</collator_enabled>
2201         <collator_strength>0</collator_strength>
2202         <presorted>N</presorted>
2203     </field>
2204 </fields>
2205 <cluster_schema/>
2206 <remotesteps>
2207     <input></input>
2208     <output></output>
2209 </remotesteps>

```

```

2210 <GUI>
2211   <xloc>1270</xloc>
2212   <yloc>750</yloc>
2213   <draw>Y</draw>
2214 </GUI>
2215 </step>
2216 <step>
2217   <name>SelectValues_Coffeeshop_1470_750</name>
2218   <type>SelectValues</type>
2219   <description/>
2220   <distributed>N</distributed>
2221   <custom_distribution/>
2222   <copies>1</copies>
2223   <partitioning>
2224     <method>none</method>
2225     <schema_name/>
2226   </partitioning>
2227   <fields>
2228     <field>
2229       <name>ID_GAME</name>
2230     </field>
2231     <field>
2232       <name>NAME</name>
2233     </field>
2234     <field>
2235       <name>MESSAGE</name>
2236     </field>
2237     <field>
2238       <name>REGION</name>
2239     </field>
2240     <field>
2241       <name>DATE</name>
2242     </field>
2243     <select_unspecified>N</select_unspecified>
2244   </fields>
2245   <cluster_schema/>
2246   <remotesteps>
2247     <input></input>
2248     <output></output>
2249   </remotesteps>
2250   <GUI>
2251     <xloc>1470</xloc>
2252     <yloc>750</yloc>
2253     <draw>Y</draw>
2254   </GUI>
2255 </step>
2256 <step>
2257   <name>SortedMerge</name>
2258   <type>SortedMerge</type>
2259   <description/>
2260   <distributed>N</distributed>
2261   <custom_distribution/>
2262   <copies>1</copies>
2263   <partitioning>
2264     <method>none</method>
2265     <schema_name/>
2266   </partitioning>
2267   <fields>
2268     <field>
2269       <name>REGION</name>
2270       <ascending>N</ascending>

```

```

2271         </field>
2272     <field>
2273         <name>ID_GAME</name>
2274         <ascending>N</ascending>
2275     </field>
2276     <field>
2277         <name>DATE</name>
2278         <ascending>N</ascending>
2279     </field>
2280 </fields>
2281 <cluster_schema/>
2282 <remotesteps>
2283     <input></input>
2284     <output></output>
2285 </remotesteps>
2286 <GUI>
2287     <xloc>1670</xloc>
2288     <yloc>250</yloc>
2289     <draw>Y</draw>
2290 </GUI>
2291 </step>
2292 <step>
2293     <name>ReplaceString_1870_250</name>
2294     <type>ReplaceString</type>
2295     <description/>
2296     <distributed>Y</distributed>
2297     <custom_distribution/>
2298     <copies>1</copies>
2299     <partitioning>
2300         <method>none</method>
2301         <schema_name/>
2302     </partitioning>
2303     <fields>
2304         <field>
2305             <in_stream_name>MESSAGE</in_stream_name>
2306             <out_stream_name>MESSAGE_WITHOUT</out_stream_name>
2307             <use_regex>yes</use_regex>
2308             <replace_string>&#x5b;&#x5e;0-9A-Za-z&#xc1;&#xc9;&#xcd;&#xd3;&#xda;&#xe1;&#xe9;&#x
                &#x5d;</replace_string>
2309             <replace_by_string/>
2310             <set_empty_string>N</set_empty_string>
2311             <replace_field_by_string/>
2312             <whole_word>no</whole_word>
2313             <case_sensitive>no</case_sensitive>
2314         </field>
2315     </fields>
2316 <cluster_schema/>
2317 <remotesteps>
2318     <input></input>
2319     <output></output>
2320 </remotesteps>
2321 <GUI>
2322     <xloc>1870</xloc>
2323     <yloc>250</yloc>
2324     <draw>Y</draw>
2325 </GUI>
2326 </step>
2327 <step>
2328     <name>ReplaceString_2070_250</name>
2329     <type>ReplaceString</type>
2330     <description/>

```

```

2331 <distributed>Y</distributed>
2332 <custom_distribution/>
2333 <copies>1</copies>
2334 <partitioning>
2335   <method>none</method>
2336   <schema_name/>
2337 </partitioning>
2338 <fields>
2339   <field>
2340     <in_stream_name>REGION</in_stream_name>
2341     <out_stream_name/>
2342     <use_regex>no</use_regex>
2343     <replace_string>02MemoryOrdenar</replace_string>
2344     <replace_by_string>Memory_HiddenRoom</replace_by_string>
2345     <set_empty_string>N</set_empty_string>
2346     <replace_field_by_string/>
2347     <whole_word>no</whole_word>
2348     <case_sensitive>no</case_sensitive>
2349   </field>
2350   <field>
2351     <in_stream_name>REGION</in_stream_name>
2352     <out_stream_name/>
2353     <use_regex>no</use_regex>
2354     <replace_string>03MemorySaturn</replace_string>
2355     <replace_by_string>Memory_Saturn</replace_by_string>
2356     <set_empty_string>N</set_empty_string>
2357     <replace_field_by_string/>
2358     <whole_word>no</whole_word>
2359     <case_sensitive>no</case_sensitive>
2360   </field>
2361   <field>
2362     <in_stream_name>REGION</in_stream_name>
2363     <out_stream_name/>
2364     <use_regex>no</use_regex>
2365     <replace_string>02OrdenarHabitacion</replace_string>
2366     <replace_by_string>HiddenRoom</replace_by_string>
2367     <set_empty_string>N</set_empty_string>
2368     <replace_field_by_string/>
2369     <whole_word>no</whole_word>
2370     <case_sensitive>no</case_sensitive>
2371   </field>
2372   <field>
2373     <in_stream_name>REGION</in_stream_name>
2374     <out_stream_name/>
2375     <use_regex>no</use_regex>
2376     <replace_string>04Saturn</replace_string>
2377     <replace_by_string>Saturn</replace_by_string>
2378     <set_empty_string>N</set_empty_string>
2379     <replace_field_by_string/>
2380     <whole_word>no</whole_word>
2381     <case_sensitive>no</case_sensitive>
2382   </field>
2383   <field>
2384     <in_stream_name>REGION</in_stream_name>
2385     <out_stream_name/>
2386     <use_regex>no</use_regex>
2387     <replace_string>08CafeteriaVersion2</replace_string>
2388     <replace_by_string>Coffeeshop</replace_by_string>
2389     <set_empty_string>N</set_empty_string>
2390     <replace_field_by_string/>
2391     <whole_word>no</whole_word>

```

```

2392         <case_sensitive>no</case_sensitive>
2393     </field>
2394 </fields>
2395 <cluster_schema/>
2396 <remotesteps>
2397     <input></input>
2398     <output></output>
2399 </remotesteps>
2400 <GUI>
2401     <xloc>2070</xloc>
2402     <yloc>250</yloc>
2403     <draw>Y</draw>
2404 </GUI>
2405 </step>
2406 <step>
2407     <name>StringOperations_2270_250</name>
2408     <type>StringOperations</type>
2409     <description/>
2410     <distributed>Y</distributed>
2411     <custom_distribution/>
2412     <copies>1</copies>
2413     <partitioning>
2414         <method>none</method>
2415         <schema_name/>
2416     </partitioning>
2417     <fields>
2418         <field>
2419             <in_stream_name>MESSAGE_WITHOUT</in_stream_name>
2420             <out_stream_name>MESSAGE_PARSED</out_stream_name>
2421             <trim_type>none</trim_type>
2422             <lower_upper>lower</lower_upper>
2423             <padding_type>none</padding_type>
2424             <pad_char/>
2425             <pad_len/>
2426             <init_cap>no</init_cap>
2427             <mask_xml>none</mask_xml>
2428             <digits>none</digits>
2429             <remove_special_characters>none</remove_special_characters>
2430         </field>
2431     </fields>
2432 <cluster_schema/>
2433 <remotesteps>
2434     <input></input>
2435     <output></output>
2436 </remotesteps>
2437 <GUI>
2438     <xloc>2270</xloc>
2439     <yloc>250</yloc>
2440     <draw>Y</draw>
2441 </GUI>
2442 </step>
2443 <step>
2444     <name>JoinRowsRegex</name>
2445     <type>JoinRows</type>
2446     <description/>
2447     <distributed>Y</distributed>
2448     <custom_distribution/>
2449     <copies>1</copies>
2450     <partitioning>
2451         <method>none</method>
2452         <schema_name/>

```

```

2453     </partitioning>
2454     <directory>&#x25;&#x25;java.io.tmpdir&#x25;&#x25;</directory>
2455         <prefix>out</prefix>
2456         <cache_size>500</cache_size>
2457         <main/>
2458         <compare>
2459             <condition>
2460                 <negated>N</negated>
2461                 <leftvalue>MESSAGE_PARSED</leftvalue>
2462                 <function>REGEXP</function>
2463                 <rightvalue>DICT</rightvalue>
2464             </condition>
2465         </compare>
2466         <cluster_schema/>
2467         <remotesteps> <input> </input> <output> </output> </remotesteps>
2468         <GUI>
2469             <xloc>2470</xloc>
2470             <yloc>250</yloc>
2471             <draw>Y</draw>
2472         </GUI>
2473     </step>
2474     <step>
2475         <name>StringOperationsDict</name>
2476         <type>StringOperations</type>
2477         <description/>
2478         <distribute>Y</distribute>
2479         <custom_distribution/>
2480         <copies>1</copies>
2481         <partitioning>
2482             <method>none</method>
2483             <schema_name/>
2484         </partitioning>
2485         <fields>
2486             <field>
2487                 <in_stream_name>NEW</in_stream_name>
2488                 <out_stream_name>DICT</out_stream_name>
2489                 <trim_type>none</trim_type>
2490                 <lower_upper>lower</lower_upper>
2491                 <padding_type>none</padding_type>
2492                 <pad_char/>
2493                 <pad_len/>
2494                 <init_cap>no</init_cap>
2495                 <mask_xml>none</mask_xml>
2496                 <digits>none</digits>
2497                 <remove_special_characters>none</remove_special_characters>
2498             </field>
2499         </fields>
2500         <cluster_schema/>
2501         <remotesteps>
2502             <input></input>
2503             <output></output>
2504         </remotesteps>
2505         <GUI>
2506             <xloc>2470</xloc>
2507             <yloc>350</yloc>
2508             <draw>Y</draw>
2509         </GUI>
2510     </step>
2511     <step>
2512         <name>TextFileInput_2470_450</name>
2513         <type>TextFileInput</type>

```

```

2514 <description/>
2515 <distributed>Y</distributed>
2516 <custom_distribution/>
2517 <copies>1</copies>
2518 <partitioning>
2519     <method>none</method>
2520     <schema_name/>
2521 </partitioning>
2522     <accept_filenames>N</accept_filenames>
2523         <passing_through_fields>N</passing_through_fields>
2524         <accept_field/>
2525         <accept_stepname/>
2526         <separator>&#x9;</separator>
2527         <enclosure/>
2528         <enclosure_breaks>N</enclosure_breaks>
2529         <escapechar>&#x5c;n</escapechar>
2530         <header>Y</header>
2531         <nr_headerlines>1</nr_headerlines>
2532         <footer>N</footer>
2533         <nr_footerlines>1</nr_footerlines>
2534         <line_wrapped>N</line_wrapped>
2535         <nr_wraps>1</nr_wraps>
2536         <layout_paged>N</layout_paged>
2537         <nr_lines_per_page>80</nr_lines_per_page>
2538         <nr_lines_doc_header>0</nr_lines_doc_header>
2539         <noempty>Y</noempty>
2540         <include>N</include>
2541         <include_field/>
2542         <rownum>N</rownum>
2543         <rownumByFile>N</rownumByFile>
2544         <rownum_field/>
2545         <format>mixed</format>
2546         <encoding>UTF-8</encoding>
2547         <add_to_result_filenames>Y</add_to_result_filenames>
2548         <file>
2549             <name>&#x24;&#x7b;Internal.Transformation.FileName.Directory&#x7d;&#x2f;..&#x
2550             <filemask/>
2551             <exclude_filemask/>
2552             <file_required>N</file_required>
2553             <include_subfolders>N</include_subfolders>
2554             <type>CSV</type>
2555             <compression>None</compression>
2556         </file>
2557         <filters>
2558         </filters>
2559         <fields>
2560             <field>
2561                 <name>NEW</name>
2562                 <type>String</type>
2563                 <format/>
2564                 <currency>&#x20ac;</currency>
2565                 <decimal>,</decimal>
2566                 <group>.</group>
2567                 <nullif>-</nullif>
2568                 <ifnull/>
2569                 <position>-1</position>
2570                 <length>6</length>
2571                 <precision>-1</precision>
2572                 <trim_type>none</trim_type>
2573                 <repeat>N</repeat>
2574             </field>

```

```

2575         </fields>
2576         <limit>0</limit>
2577         <error_ignored>N</error_ignored>
2578         <skip_bad_files>N</skip_bad_files>
2579         <file_error_field/>
2580         <file_error_message_field/>
2581         <error_line_skipped>N</error_line_skipped>
2582         <error_count_field/>
2583         <error_fields_field/>
2584         <error_text_field/>
2585         <bad_line_files_destination_directory/>
2586         <bad_line_files_extension>warning</bad_line_files_extension>
2587         <error_line_files_destination_directory/>
2588         <error_line_files_extension>error</error_line_files_extension>
2589         <line_number_files_destination_directory/>
2590         <line_number_files_extension>line</line_number_files_extension>
2591         <date_format_lenient>Y</date_format_lenient>
2592         <date_format_locale>es_ES</date_format_locale>
2593         <shortFileName/>
2594         <pathFieldName/>
2595         <hiddenFieldName/>
2596         <lastModificationTimeFieldName/>
2597         <uriNameFieldName/>
2598         <rootUriNameFieldName/>
2599         <extensionFieldName/>
2600         <sizeFieldName/>
2601         <cluster_schema/>
2602         <remotesteps>
2603             <input></input>
2604             <output></output>
2605         </remotesteps>
2606         <GUI>
2607             <xloc>2470</xloc>
2608             <yloc>450</yloc>
2609             <draw>Y</draw>
2610         </GUI>
2611     </step>
2612     <step>
2613         <name>UniqueRowsByHashSet_2670_250</name>
2614         <type>UniqueRowsByHashSet</type>
2615         <description/>
2616         <distributed>Y</distributed>
2617         <custom_distribution/>
2618         <copies>1</copies>
2619         <partitioning>
2620             <method>none</method>
2621             <schema_name/>
2622         </partitioning>
2623         <store_values>N</store_values>
2624         <reject_duplicate_row>N</reject_duplicate_row>
2625         <error_description/>
2626         <fields>
2627             <field>
2628                 <name>REGION</name>
2629             </field>
2630             <field>
2631                 <name>NAME</name>
2632             </field>
2633             <field>
2634                 <name>DATE</name>
2635             </field>

```



```

2636         </fields>
2637         <cluster_schema/>
2638         <remotesteps>
2639             <input></input>
2640             <output></output>
2641         </remotesteps>
2642         <GUI>
2643             <xloc>2670</xloc>
2644             <yloc>250</yloc>
2645             <draw>Y</draw>
2646         </GUI>
2647     </step>
2648     <step>
2649         <name>ejemplo</name>
2650         <type>TextFileOutput</type>
2651         <description/>
2652         <distributed>Y</distributed>
2653         <custom_distribution/>
2654         <copies>1</copies>
2655         <partitioning>
2656             <method>none</method>
2657             <schema_name/>
2658         </partitioning>
2659         <separator>&#x3b;</separator>
2660     <enclosure>&#x22;</enclosure>
2661     <enclosure_forced>N</enclosure_forced>
2662     <enclosure_fix_disabled>N</enclosure_fix_disabled>
2663     <header>Y</header>
2664     <footer>N</footer>
2665     <format>DOS</format>
2666     <compression>None</compression>
2667     <encoding>UTF-8</encoding>
2668     <endedLine/>
2669     <fileNameInField>N</fileNameInField>
2670     <fileNameField/>
2671     <create_parent_folder>Y</create_parent_folder>
2672     <file>
2673         <name>&#x24;&#x7b;Internal.Transformation.FileName.Directory&#x7d;&#x2f;..&#x2f;output&#x2f;e
2674         <is_command>N</is_command>
2675         <servlet_output>N</servlet_output>
2676         <do_not_open_new_file_init>N</do_not_open_new_file_init>
2677         <extention>csv</extention>
2678         <append>N</append>
2679         <split>N</split>
2680         <haspartno>N</haspartno>
2681         <add_date>N</add_date>
2682         <add_time>N</add_time>
2683         <SpecifyFormat>N</SpecifyFormat>
2684         <date_time_format/>
2685         <add_to_result_filenames>Y</add_to_result_filenames>
2686         <pad>N</pad>
2687         <fast_dump>N</fast_dump>
2688         <splitevery>0</splitevery>
2689     </file>
2690 </fields>
2691
2692     <field>
2693         <name>ID_GAME</name>
2694         <type>Integer</type>
2695         <format/>
2696         <currency/>

```

```

2697         <decimal>,</decimal>
2698     </group>.</group>
2699     <nullif/>
2700     <trim_type>none</trim_type>
2701     <length>-1</length>
2702     <precision>0</precision>
2703 </field>
2704 <field>
2705     <name>NAME</name>
2706     <type>String</type>
2707     <format/>
2708     <currency/>
2709     <decimal/>
2710     <group/>
2711     <nullif/>
2712     <trim_type>none</trim_type>
2713     <length>-1</length>
2714     <precision>-1</precision>
2715 </field>
2716 <field>
2717     <name>REGION</name>
2718     <type>String</type>
2719     <format/>
2720     <currency/>
2721     <decimal/>
2722     <group/>
2723     <nullif/>
2724     <trim_type>none</trim_type>
2725     <length>-1</length>
2726     <precision>-1</precision>
2727 </field>
2728 <field>
2729     <name>DATE</name>
2730     <type>Timestamp</type>
2731     <format>yyyy&#x2f;MM&#x2f;dd HH&#x3a;mm&#x3a;ss</format>
2732     <currency/>
2733     <decimal/>
2734     <group/>
2735     <nullif/>
2736     <trim_type>none</trim_type>
2737     <length>400</length>
2738     <precision>-1</precision>
2739 </field>
2740 <field>
2741     <name>MESSAGE</name>
2742     <type>String</type>
2743     <format/>
2744     <currency/>
2745     <decimal/>
2746     <group/>
2747     <nullif/>
2748     <trim_type>none</trim_type>
2749     <length>-1</length>
2750     <precision>-1</precision>
2751 </field>
2752 </fields>
2753 <cluster_schema/>
2754 <remotesteps>
2755     <input></input>
2756     <output></output>
2757 </remotesteps>

```

```
2758 <GUI>
2759   <xloc>2870</xloc>
2760   <yloc>250</yloc>
2761   <draw>Y</draw>
2762 </GUI>
2763 </step>
2764   <step_error_handling>
2765   </step_error_handling>
2766   <slave-step-copy-partition-distribution>
2767   </slave-step-copy-partition-distribution>
2768   <slave_transformation>N</slave_transformation>
2769 </transformation>
```


Apéndice E

Archivo metamodelo Ecore

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ecore:EPackage xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="modelo"
   nsURI="http://www.eclipse.org/vwql/modelo/model" nsPrefix="modelo">
4   <eClassifiers xsi:type="ecore:EClass" name="Evidence">
5     <eStructuralFeatures xsi:type="ecore:EReference" name="target" lowerBound="1"
6       eType="//Target" containment="true"/>
7     <eStructuralFeatures xsi:type="ecore:EReference" name="outputs" upperBound="-1"
8       eType="//Output" containment="true"/>
9     <eStructuralFeatures xsi:type="ecore:EReference" name="indications"
10      eType="//Indication"
11      containment="true"/>
12     <eStructuralFeatures xsi:type="ecore:EAttribute" name="name"
13       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EString"/>
14     <eStructuralFeatures xsi:type="ecore:EReference" name="entities" upperBound="-1"
15       eType="//Entity" containment="true"/>
16   </eClassifiers>
17   <eClassifiers xsi:type="ecore:EClass" name="Target">
18     <eStructuralFeatures xsi:type="ecore:EReference" name="players" upperBound="-1"
19       eType="//Player" containment="true"/>
20     <eStructuralFeatures xsi:type="ecore:EAttribute" name="operator"
21       eType="//Operator"/>
22   </eClassifiers>
23   <eClassifiers xsi:type="ecore:EClass" name="Player">
24     <eStructuralFeatures xsi:type="ecore:EAttribute" name="id"
25       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EString"/>
26   </eClassifiers>
27   <eClassifiers xsi:type="ecore:EClass" name="Output">
28     <eStructuralFeatures xsi:type="ecore:EAttribute" name="format"
29       eType="//Format"/>
30   </eClassifiers>
31   <eClassifiers xsi:type="ecore:EClass" name="Entity">
32     <eStructuralFeatures xsi:type="ecore:EAttribute" name="activity"
33       eType="//Activity"/>
34   </eClassifiers>
35   <eClassifiers xsi:type="ecore:EClass" name="Indication">
36     <eStructuralFeatures xsi:type="ecore:EAttribute" name="type" eType="//Type"/>
37     <eStructuralFeatures xsi:type="ecore:EAttribute" name="date_beg"
38       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EDate"/>
39     <eStructuralFeatures xsi:type="ecore:EAttribute" name="date_end"
40       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EDate"/>
41   </eClassifiers>
```

```

34 <eClassifiers xsi:type="ecore:EClass" name="Memory" eSuperTypes="#//Indication">
35   <eStructuralFeatures xsi:type="ecore:EAttribute" name="type_specific"
      eType="#//TypeMemory"/>
36 </eClassifiers>
37 <eClassifiers xsi:type="ecore:EClass" name="Hidden_Room"
      eSuperTypes="#//Indication">
38   <eStructuralFeatures xsi:type="ecore:EAttribute" name="type_specific"
      eType="#//TypeHiddenRoom"/>
39 </eClassifiers>
40 <eClassifiers xsi:type="ecore:EClass" name="CoffeeShop"
      eSuperTypes="#//Indication">
41   <eStructuralFeatures xsi:type="ecore:EAttribute" name="type_specific"
      eType="#//TypeCoffeeShop"/>
42 </eClassifiers>
43 <eClassifiers xsi:type="ecore:EClass" name="Saturn" eSuperTypes="#//Indication">
44   <eStructuralFeatures xsi:type="ecore:EAttribute" name="type_specific"
      eType="#//TypeSaturn"/>
45 </eClassifiers>
46 <eClassifiers xsi:type="ecore:EEnum" name="Operator">
47   <eLiterals name="get"/>
48   <eLiterals name="count" value="1"/>
49 </eClassifiers>
50 <eClassifiers xsi:type="ecore:EEnum" name="Format">
51   <eLiterals name="csv"/>
52 </eClassifiers>
53 <eClassifiers xsi:type="ecore:EEnum" name="Type">
54   <eLiterals name="words"/>
55   <eLiterals name="sentences" value="1"/>
56   <eLiterals name="single" value="2"/>
57   <eLiterals name="turns" value="3"/>
58   <eLiterals name="time" value="4"/>
59   <eLiterals name="points" value="5"/>
60   <eLiterals name="greetings" value="7"/>
61   <eLiterals name="checks" value="8"/>
62   <eLiterals name="WHquestions" value="6"/>
63   <eLiterals name="exclamations" value="9"/>
64   <eLiterals name="game" value="10"/>
65   <eLiterals name="words_german" value="11"/>
66   <eLiterals name="emojis" value="12"/>
67   <eLiterals name="prepositions" value="13"/>
68   <eLiterals name="confirmations" value="14"/>
69   <eLiterals name="communication_strategies" value="15"/>
70 </eClassifiers>
71 <eClassifiers xsi:type="ecore:EEnum" name="TypeMemory">
72   <eLiterals name="panel"/>
73   <eLiterals name="panel_points" value="1"/>
74 </eClassifiers>
75 <eClassifiers xsi:type="ecore:EEnum" name="TypeHiddenRoom">
76   <eLiterals name="distance" value="1"/>
77 </eClassifiers>
78 <eClassifiers xsi:type="ecore:EEnum" name="TypeCoffeeShop">
79   <eLiterals name="distance"/>
80   <eLiterals name="clients" value="1"/>
81   <eLiterals name="client_points" value="2"/>
82 </eClassifiers>
83 <eClassifiers xsi:type="ecore:EEnum" name="TypeSaturn">
84   <eLiterals name="role"/>
85 </eClassifiers>
86 <eClassifiers xsi:type="ecore:EEnum" name="Activity">
87   <eLiterals name="Memory_Saturn"/>
88   <eLiterals name="Memory_HiddenRoom" value="1"/>

```

```
89     <eLiterals name="HiddenRoom" value="2"/>
90     <eLiterals name="Saturn" value="3"/>
91     <eLiterals name="Mango" value="4"/>
92     <eLiterals name="Coffeeshop" value="5"/>
93 </eClassifiers>
94 </ecore:EPackage>
```


Apéndice F

Publicaciones

F.1. Publicación enviada a TEEM'15

Proceedings
TEEM'15

Third International Conference on Technological Ecosystems for Enhancing Multiculturality

Porto, Portugal
October 7th – 9th, 2015

Editors:

Gustavo R. Alves
Polytechnic of Porto

Manuel C. Felgueiras
Polytechnic of Porto

TEEM'15 is jointly organized by the Centre for Innovation in Engineering and Industrial Technology at the Polytechnic of Porto – School of Engineering and the Research GRoup in InterAction and eLearning (GRIAL) and Research Institute for Educational Sciences (IUCE) at the University of Salamanca.



Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Designed by:

Felicidad García Sánchez
Research GRoup in InterAction and eLearning (GRIAL)

José Silva
Polytechnic of Porto – School of Engineering (ISEP)

The Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York New York 10121-0701

ISBN: 978-1-4503-3442-6

Doctoral Consortium	617
<i>Doctoral Consortium</i>	
Francisco José García-Peñalvo	619
<i>Demographic Change – Towards a framework to manage IT-personnel in times of scarcity of talent</i>	
Olaf Radant, Ricardo Colomo-Palacios and Vladimir Stantchev	623
<i>Evolutionary Analysis of Online Labs</i>	
Raúl Cordeiro, José Manuel Fonseca and Gustavo R. Alves	631
<i>The audiovisual and digital media as transmitting vehicle between academic world and work world in higher education</i>	
David Martínez Sánchez, Francisco Javier Frutos Esteban and Isabel M. Barrios Vicente	639
<i>Personal Learning Environments (PLE) in The Teaching Ecosystem: A proposal in development</i>	
Cecilia Aranda Cirínez and Ana García-Valcárcel	645
<i>A Federation of Online Labs for Assisting Engineering and Sciences Education in the MENA Region</i>	
Razwan Mohmed Salah, Pedro Guerreiro and Gustavo R. Alves	651
6th International Workshop on Software Engineering for E-learning (ISELEAR15)	659
<i>6th International Workshop on Software Engineering for E-learning (ISELEAR15)</i>	
Miguel Ángel Conde-González, José Luis Sierra Rodríguez and Antonio Sarasa-Cabezuelo	661
<i>Definition and deployment of a non-formal learning environment in a business context</i>	
Miguel Á. Conde, Gonzalo Esteban-Costales, Victor Rodríguez-Mendez, Camino Fernández-Llamas and Luis A. Amador	667
<i>A Domain Specific Language to retrieve objective indicators for foreign language learning in virtual worlds</i>	
Antonio Balderas, Anke Berns, Manuel Palomo-Duarte, Juan M. Dodero, Raúl Gómez-Sánchez and Iván Ruiz-Rube	675
<i>Influence of the students' learning strategy on the evaluation scores</i>	
P. Molins-Ruano, F. Jurado, P. Rodríguez, S. Atrio and G. M. Sacha	681
<i>Faat – Freelance as a Team</i>	
Rodrigo Borrego Bernabé, Iván Álvarez Navia and Francisco José García-Peñalvo	687
<i>Definition of a Technological Ecosystem for Scientific Knowledge Management in a PhD Programme</i>	
Alicia García-Holgado, Francisco J. García-Peñalvo and María José Rodríguez-Conde	695
Authors Index	701

A Domain Specific Language to retrieve objective indicators for foreign language learning in virtual worlds

Antonio Balderas University of Cadiz Spain, Puerto Real, Av. Universidad de Cádiz 10, CP 11519 +34 956 48 34 80 antonio.balderas@uca.es	Anke Berns University of Cadiz Spain, Cádiz, Av. Gómez Ulla s/n, CP 11003 +34 956 01 55 49 anke.berns@uca.es	Manuel Palomo-Duarte University of Cadiz Spain, Puerto Real, Av. Universidad de Cádiz 10, CP 11519 +34 956 48 34 36 manuel.palomo@uca.es
Juan M. Dodero University of Cadiz Spain, Puerto Real, Av. Universidad de Cádiz 10, CP 11519 +34 956 48 34 35 juanma.dodero@uca.es	Raúl Gómez-Sánchez University of Cadiz Spain, Puerto Real, Av. Universidad de Cádiz 10, CP 11519 +34 956 48 32 00 raul.gomezsa@alum.uca.es	Iván Ruiz-Rube University of Cadiz Spain, Puerto Real, Av. Universidad de Cádiz 10, CP 11519 +34 956 48 34 85 ivan.ruiz@uca.es

ABSTRACT

Virtual worlds (VWs) have become increasingly popular to support students' foreign language learning, especially beyond the classroom. Unfortunately students' interaction in VWs is not always available for the supervisor and thus is not easy to analyse. Nonetheless, it provides interesting information not only in terms of assessment, but also to detect general learner profiles and trends in their use. Therefore we propose a computer system that accepts queries in a simple but specific language for VWs, allowing the supervisor to design assessment formulas, generate reports on students' interaction in VWs and refine the formulas from these results until obtaining a valid indicator for analysing and assessing the retrieved data. In this paper, we describe a case study which has been carried out in a German foreign language course, using a VW implemented video-game and assessing students' foreign language skills. The study has been deployed in a scalable way through our computer system. Preliminary results provide a set of indicators for individual and group behaviour which can be used to assess student's ability to communicate in the target language.

Categories and Subject Descriptors

K.3.1 [Computers and Education]: Computer Uses in Education- *Computer-assisted instruction (CAI)*

General Terms

Measurement, Design, Experimentation, Languages.

Keywords

virtual worlds, computer-supported foreign language learning, learning analytics, domain specific language, assessment

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. TEEM '15, October 07 - 09, 2015, Porto, Portugal Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-3442-6/15/10...\$15.00 DOI: <http://dx.doi.org/10.1145/2808580.2808683>

1. INTRODUCTION

With the increasing rise of Information and Communications Technologies (ICTs) and the use of blended teaching practices teachers and academic institutions have started exploring the potential of different kinds of virtual learning environments (VLEs) to support their students' foreign language learning. The main benefits of VLEs consist in providing its users with access to learning contents, services and applications anytime, anyplace and at any pace. Most widely used VLEs are Learning Management Systems (LMSs), virtual worlds (VWs) and Social Network Sites (SNS). Due to their huge variety of tools (chat, forums, etc.) they offer teachers the opportunity to create highly interactive learning environments to be accessed even outside the class [3]. Thus interaction in and with the target language is no longer restricted to the classroom but can easily go beyond, bridging the gap between in- and out-of-classroom learning. In recent years there has been furthermore an increasing attempt to explore the educational potential of computer games to engage students in foreign language learning, especially beyond the class setting [7]. This has been done both through the adaptation of a variety of commercial games (World of Warcraft, The Sims, Wonderland, Lineage II, etc.) [23] as well as through the specific design of educational games, also called serious games [4]. Even though many researchers have recognised the great educational and motivational potential of computer games there are still few empirical studies which have directly investigated their impact on students' learning [2]. The same can be said of learning environments such as VWs (Second Life, OpenCobalt, etc.) [15]. Some of the reasons for this are undoubtedly related to the fact that neither commercial video games nor most of the VWs are open-source software and thus do not allow teachers to design their own learning environment or to trace back students' interactions [8]. This makes it extremely difficult, if not impossible, to analyse students' interaction and its impact on their learning progress as well as learning outcomes [21]. According to [14], the aforementioned data offer valuable information to be used for the assessment and analysis of students' interaction in VWs. These and other reasons encouraged us to design firstly our own VW game-environment based on OpenSim (an open source-software) and secondly, to create a Domain Specific Language (DSL) to retrieve objective indicators for assessing foreign language learning in VWs.



Figure 1: Players performing the Saturn game

For this purpose we have defined VWQL, a generative computer language to customise language learning analytics in a VW. It is a formal language that can express simple written queries using a simple syntax and a domain-specific vocabulary. This way, educators can easily retrieve indicators from the information stored in the database without requiring deep technical skills on databases or computer programming. We have developed EvalSim, a program that automatically process queries written in VWQL to obtain the desired indicators from a large number of students or games, being a scalable approach. These queries, which express the design assessment formulas, can be refined until obtaining a valid indicator for analysing and assessing students' behaviour. In the following we present a case study, where EvalSim retrieves information from Saturn, an OpenSim-based VW game, which initially has been designed for German language learning, but which can easily be used for assessing other OpenSim based video games.

The rest of this paper sets out the following sections as so: the second section describes the theoretical background both, for the use of computer games in foreign language learning as well as for the use of DSL. The third section introduces our methodology and experimental settings, followed by the fourth section which will show the conducted case study and results obtained. Finally, in the last section, we will provide a discussion along with some preliminary conclusions and future research lines.

2. THEORETICAL BACKGROUND

2.1 Game-based learning and foreign language acquisition

Taking into consideration that the development of a second language is almost wholly dependent on the amount of comprehensible input and output a learner receives and produces [4], several linguists have stressed the importance of the following aspects: to provide foreign language learners with those kinds of interactions that facilitate the intake of comprehensible input [19, 28, 17], stimulate the production of language output [26], encourage the negotiation of meaning [22] and facilitate noticing [25]. Due to Swain *output production* is key both to enhance learners' fluency in the target language as well as to make them aware of their language deficits. As stressed by Swain [26] *pushed output* is an important factor to focus learners' attention on aspects such as form and feedback. The positive and negative feedback learners get during their interaction with others is seen as a key-issue for foreign language acquisition, since it provides learners with the possibility to modify their language output and therefore to develop their foreign language proficiency [19]. The above mentioned aspects make clear why versatile interaction with other speakers of the target language, independently if these are native or non-native

speakers, becomes a key issue for a successful foreign language acquisition [4]. Keeping in mind the aforementioned aspects the current project aims to explore the potential of 3D virtual online games to reinforce students' foreign language learning through versatile interaction. Furthermore by designing our own virtual game environment and DSL the current study intends to offer some example on how to assess students' learning outcomes when using VWs.

2.2 Domain-specific languages

Model Driven Development (MDD) is a software development approach that has gained importance during the last years. It is based on generators that automatically create the code implementing the system functionalities. Thus, application-level development focus shifts from manual coding to modeling [18]. DSL generation is one of the products of a MDD process [10]. DSL are languages that support the thinking in terms of domain, rather than the thinking in terms of the *representation language* [9].

We can find many works in the literature that implement DSL to support the management of their models: to specify and constrain synthetic biological systems [5], to manage plans for composite applications [16], to manage complex event processing [6], to control integration of life sciences data [11], to verify dialectical games [27], to initialise wiki contents [12] and so forth.

Moreover we can find educational works which equally address their models through DSL. In [1] was introduced SASQL, a DSL, to obtain objective indicators from students' interaction with a Moodle LMS regarding their performance in several generic skills such as teamwork, planning and time management, interpersonal skills and leadership. Additionally, in [20] a DSL has been developed for the generation of LMS modules.

In our proposal the DSL obtains indicators from students' interaction with the VW game environment. Analysing students' game behaviour might help us to assess students' learning outcomes as well as to improve the game design itself.

3. METHODOLOGY AND EXPERIMENTAL SETTINGS

The VW we have designed for the present study was based on a 3-D virtual role-play, called Saturn¹, in which students were asked to perform a joint shopping task. Each team compounded of two players, each one of which took a different role: Whilst player A was asked to perform as a client, player B took the role of the shop-assistant. To play the game successfully both players must coordinate their actions communicating via text-chat. In figure 1 (on the left) we can observe one of the participating students performing as a client and another student (on the right) performing as a shop-assistant.

¹ Available at <https://code.google.com/p/daifceale>

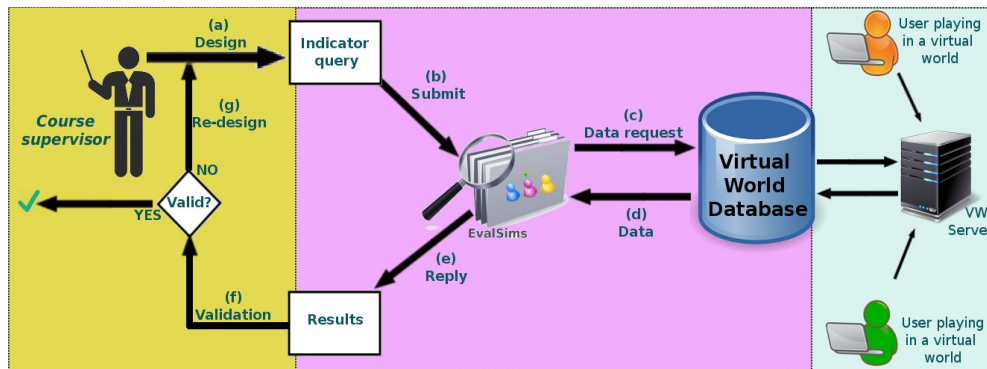


Figure 2: Hypothesis contrast cycle

Before explaining our case study, we will introduce the DSL we have used for the current case study and which is called VWQL. VWQL is the domain-specific language we have defined to retrieve objective indicators in OpenSim VWs. EvalSim is a system to process VWQL queries. It follows a model-driven approach [24] for modeling procedures to obtain the requested indicators. EvalSim has been created using Xtext [13] within the Eclipse Modeling Framework. It is provided as open-source software under the terms of the GNU General Public License.

As we can observe in figure 2, we first design (a) and submit (b) a query to request information about the concrete indicators, we would like to obtain, then EvalSim extracts the necessary information from the VW database (c, d) to meet our request for indicators. This information is then processed and returns (e) the raw data into an XML file that can be processed with an analytics or spreadsheet software. If we consider that the information provided is not valid (f) as an indicator or might need to be fine-tuned, we can redesign (g) the query and restart the process.

Language syntax (beta version 0.1) and reserved words are illustrated in listing 1. The first line starts with the compulsory term **Evidence** and the name of the evidence or indicator. It will be used to identify the different files produced by EvalSim. The second line starts with the compulsory words **get students**. Furthermore the second line indicates, if we want to obtain information about all students involved in the game-experience or only about some of the them. In the last case we need to specify students' identifiers after the reserved words. Finally, the last line specifies the kind of information to be retrieved. This information is being indicated after the compulsory term **show** and can be one or more of the following:

- **words** Number of words written in the text-chat. By default it counts all words, even though the name of a specific dictionary could be provided in order to solely count the words contained in it.
- **sentences** Number of sentences written in the text-chat.
- **single** Number of single-word sentences written in the text-chat.
- **turns** Number of turns taken in the text-chat. A turn is a set of consecutive phrases written by the same player.
- **time** Number of minutes played in the virtual world.

- **points** Number of points obtained in the virtual world. The ways to obtain points depend on the specific virtual world that had been played.

Listing 1: Reserved words and syntax of VWQL (version 0.1)

```
Evidence name_of_the_evidence:
get students [id_of_the_student]
show ( words [dict] | sentences | single |
| turns | time | points )+
```

Initially, the VWQL included the indicators we used for the assessment process in a previous case study with a similar setting [4]. Later we included the indicator of points to accomplish the requirements of score-based assessment case studies, like the one we have conducted in [3]. Finally, the indicator set was extended by including also a time indicator, which we considered another interesting indicator to be employed for the current case study.

4. CASE STUDY

The case study we analyse in the following was developed at a Spanish high school in a compulsory German foreign language course from the B1 Common European Framework of Reference (CEFR) level. The experiment has been carried out with 5 students who were studying in their eleventh grade and who played several games in the VW. In the following we will describe two examples of information queries we have launched during the experience.

In order to detect students' ability to communicate in a foreign language, the following hypothesis was established: a student had difficulties to make himself understood if he needed two or more sentences, per turn, to communicate with his teammate.

In line with the aforementioned hypothesis we firstly considered the speaking pace as an interesting indicator, comparing the amount of time each student played with the amount of sentences written by him. To obtain this indicator we designed this first formula (listing 2).

Listing 2: First group query

```
Evidence time_sentences:
get students
show time , sentences
```

In table 1, we can observe that almost all participating students used writing an average of 1 sentence per minute. Thus only one student (Student4) wrote 2.22 sentences per minute. Moreover, in the chart (figure 3), we can see that Student4 wrote much more sentences in the text-chat compared to the rest of his classmates (red column). Nonetheless, Student4 did not play longer than his classmates (blue column).

Table 1. Result of the first group query

Student	Speaking pace
Student1	1.22
Student2	1.50
Student3	0.80
Student4	2.22
Student5	1.50

This suggested that Student4 probably encountered difficulties in communicating fluently with his playmate. Therefore he probably had to re-write and clarify many sentences to successfully perform the game.

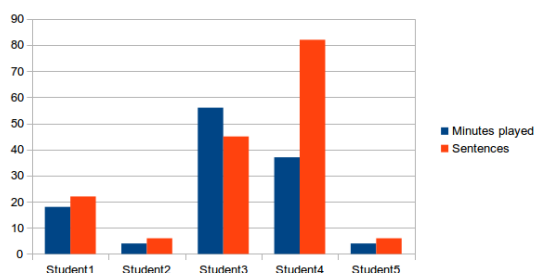


Figure 3: Chart showing minutes played and sentences written

However, a difference of one sentence per minute was not a significant difference. Hence, we decided to redesign the formula considering this time, the length of the turns each student played, rather than the amount of phrases written by him. The length of a turn refers to the number of phrases written by a transmitter student before the receiver student acts and answers. In this case the formula is represented in listing 3.

Listing 3: Second group query

```
Evidence sentences_turns:
get students
show sentences , turns
```

The chart (figure 4) and table 2 confirm that to make himself understood Student4 needed to write twice the phrases (per turn) the rest of the participating students needed to perform the same game task. Therefore this could be an objective indicator for the fact that Student4 had greater difficulties than other students when communicating in the target language via the VW text-chat.

Table 2. Result of the second group query

Student	Average of sentences per turn
Student1	1.83
Student2	2.00
Student3	2.14
Student4	4.10
Student5	2.00

With this second query, the course supervisor obtained objective data to demonstrate that Student4 had a lower communication level in German language than his playmate.

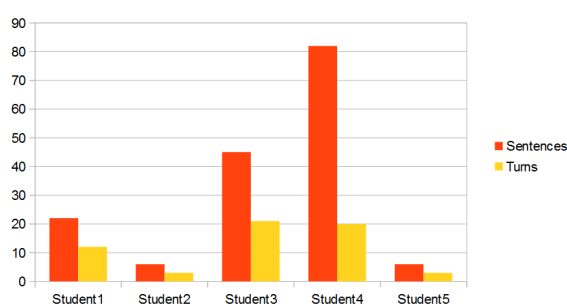


Figure 4: Chart showing turns played and sentences written

This confirms that our first hypothesis was correct, which is also reinforced by the fact that the other pairs of the columns are balanced (in the here analysed games students usually wrote approximately 1 sentence per minute played). However, compared to his classmates, Student4 took less time to write much more sentences. This suggests that Student4 probably encountered difficulties when communicating via text-chat with his playmate. This led us to the hypothesis that he probably had to re-write and clarify many sentences to achieve the same goal in the game than the rest of mates. Hence the ratio of phrases per turn (as obtained by the query 3) could be an objective indicator for assessing students' ability to *make themselves understood in the given context (the German foreign language course - B1)*. In particular, Student4 had a poor performance needing more than two sentences per turn to communicate.

5. CONCLUSIONS

VWs have become increasingly popular in foreign language learning. They support students' learning while offering new opportunities to design attractive game-applications, engaging students' interaction with and in the target language. Furthermore, a course supervisor could be provided with interesting information on students' interaction in VWs. This information is interesting not only in terms of assessment, but also to detect general learner profiles and trends in the use of VWs. However, usually this information is not available for the supervisor and, even if it were, it is not easy to analyse.

We here propose using VWQL, a DSL to obtain the required information from students' interaction with the VW game-environment. The VWQL syntax is very simple, so even educators with limited computer programming or database management competences can easily obtain very useful

information from the VW logs. After writing queries in the simple language, a report with the data requested will be automatically generated by EvalSim. This way, we are able to analyse students' game behaviour in a scalable way improving the learning as well as VW game-design.

In the present case study we have described the features of the first beta version of EvalSim. The results obtained are promising. Although it is only a first approximation, the interpretation, we did in our case study of the indicators, helped us to build a hypothesis that was contrasted in a second query. Hence the current study provides an example of how using our system (EvalSim) the course supervisor can easily propose and refine formulas to assess students' individual performance in different skills, justifying it with objective figures. However, the results of this case study are still limited, and need further study to draw a stronger conclusion on the validity of the specific usage of the information retrieved by VWQL.

In future works we aim to enhance the above described model to obtain a greater variety of data such as Wh-questions, exclamations and other interesting strategies, students might use while performing the game. This way, we could make a stronger validation of our proposal in a case study with a greater sample size.

6. ACKNOWLEDGMENTS

This work has been funded by the European Union under the OpenDiscoverySpace (CIP-ICT-PSP-2011-5) and UBICAMP (526843_LLP-1-2012 Es-Erasmus-ESMO) projects.

7. REFERENCES

- [1] Balderas, A., Ruiz-Rube, I., Palomo-Duarte, M., Dodero, J.M.: A generative computer language to customize online learning assessments. In: *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. pp. 141–147. ACM (2013)
- [2] Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., Berta, R.: Assessment in and of serious games: An overview. *Advances in Human-Computer Interaction* 2013, 1:1–1 (Jan 2013)
- [3] Berns, A., Gonzalez-Pardo, A., Camacho, D.: Game-like language learning in 3-D virtual environments. *Computers & Education* 60(1), 210 – 220 (2013)
- [4] Berns, A., Palomo-Duarte, M., Dodero, J.M., Valero-Franco, C.: Using a 3D online game to assess students' foreign language acquisition and communicative competence. In: Leo, D.H., Ley, T., Klamma, R., Harrer, A. (eds.) *Proceedings of the 8th European Conference, on Technology Enhanced Learning (EC-TEL)*. Lecture Notes in Computer Science, vol. 8095, pp. 19–31. Springer (2013)
- [5] Bilitchenko, L., Liu, A., Cheung, S., Weeding, E., Xia, B., Leguia, M., Anderson, J.C., Densmore, D.: Eugene—a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PloS one* 6(4), e18882 (2011)
- [6] Boubeta-Puig, J., Ortiz, G., Medina-Bulo, I.: Model4cep: Graphical domain-specific modeling languages for CEP domains and event patterns. *Expert Systems with Applications* 42(21), 8095 – 8110 (2015)
- [7] Brom, C., Preuss, M., Klement, D.: Are educational computer micro-games engaging and effective for knowledge acquisition at high-schools? A quasi-experimental study. *Computers & Education* 57(3), 1971 – 1988 (2011)
- [8] Cruz-Benito, J., Therón, R., García-Peñalvo, F.J., Lucas, E.P.: Discovering usage behaviors and engagement in an educational virtual world. *Computers in Human Behavior* 47(0), 18 – 25 (2015)
- [9] Decker, S.: On domain-specific declarative knowledge representation and database languages. In: *KRDB-98 Proceedings of the 5th Workshop Knowledge Representation meets DataBases*. Seattle, WA (1998)
- [10] van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. *SIGPLAN Not.* 35(6), 26–36 (Jun 2000)
- [11] Deus, H.F., Correa, M.C., Stanislaus, R., Miragaia, M., Maass, W., De Lencastre, H., Fox, R., Almeida, J.S.: S3ql: A distributed domain specific language for controlled semantic integration of life sciences data. *BMC bioinformatics* 12(1), 285 (2011)
- [12] Díaz, O., Puente, G.: A DSL for corporate wiki initialization. In: Mouratidis, H., Rolland, C. (eds.) *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, vol. 6741, pp. 237–251. Springer Berlin Heidelberg (2011)
- [13] Eysholdt, M., Behrens, H.: Xtext - implement your language faster than the quick and dirty way tutorial summary. In: *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, SPLASH '10*. pp. 307–309 (2010)
- [14] Fidalgo-Blanco, Á., Sein-Echaluze, M.L., García-Peñalvo, F.J., Conde, M.Á.: Using learning analytics to improve teamwork assessment. *Computers in Human Behavior* 47(0), 149 – 156 (2015)
- [15] Hew, K.F., Cheung, W.S.: Use of three-dimensional (3-D) immersive virtual worlds in K-12 and higher education settings: A review of the research. *British Journal of Educational Technology* 41(1), 33–55 (2010)
- [16] Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: BPMN4TOSCA: A domain-specific language to model management plans for composite applications. Springer (2012)
- [17] Krashen, S.: *The input hypothesis*. Longman, New York (1985)
- [18] Liggesmeyer, P., Trapp, M.: Trends in embedded software engineering. *Software, IEEE* 26(3), 19–25 (2009)
- [19] Long, M.: The role of the linguistic environment in second language acquisition. In: Ritchie, W., Bhatia, T. (eds.) *Handbook of Research on Language Acquisition*. Second Language Acquisition, vol. 2, pp. 413–468. Academic, New York (1996)
- [20] Montenegro-Marín, C.E., Cueva-Lovelle, J.M., Sanjuan-Martinez, O., García-Díaz, V.: Domain specific language for the generation of learning management systems modules. *Journal of Web Engineering* 11(1), 23–50 (2012)
- [21] Moreno-Ger, P., Martínez-Ortiz, I., Freire, M., Manero, B., Fernández-Manjon, B.: Serious games: A journey from research to application. In: *IEEE Frontiers in Education Conference* (2014)
- [22] Pica, T.: Research on negotiation: What does it reveal about second-language learning conditions, processes, and outcomes? *Language Learning* 44(3), 493–527 (1994)

- [23] Reinders, H., Wattana, S.: Learn english or die: The effects of digital games on interaction and willingness to communicate in a foreign language. *Digital Culture & Education* 3, 3–29 (2011)
- [24] Schmidt, D.C.: Model-driven engineering. *Computer* 39(2), 25–31 (2006)
- [25] Schmidt, R.W.: The role of consciousness in second language learning. *Applied Linguistics* 11(2), 129–158 (1990)
- [26] Swain, M.: The output hypothesis: Theory and research. In: Hinkel, E. (ed.) *Handbook on Research in Second Language Teaching and Learning*, pp. 471–484. Lawrence Erlbaum, Mahwah (2005)
- [27] Wells, S., Reed, C.: A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic* 10(4), 309–329 (2012)
- [28] Ziegler, N., Seals, C., Ammons, S., Lake, J., Hamrick, P., Rebuschat, P.: Interaction in conversation groups. In: McDonough, K., Mackey, A. (eds.) *Second Language Interaction in Diverse Educational Contexts*, chap. 14. John Benjamins Publishing Company (2013)

F.2. Publicación enviada a JID2016

Jornadas de Innovación Docente Universitaria UCA

VWQL: un lenguaje específico de dominio para la obtención de indicadores objetivos del aprendizaje de idiomas en mundos virtuales

Antonio Balderas Alberico*, Manuel Palomo Duarte*, Juan Manuel Dodero Beardo*, Anke Berns†, Raúl Gómez Sánchez*, Iván Ruiz Rube*

*Departamento de Ingeniería Informática, Escuela Superior de Ingeniería, †Departamento de Filología Francesa e Inglesa, Facultad de Filosofía y Letras

antonio.balderas@uca.es

RESUMEN: El uso de mundos virtuales en la enseñanza se ha extendido en los últimos años. La descontextualización en que se sumergen los alumnos y la capacidad de registro de las interacciones que se producen (movimientos, intercambio de mensajes, acciones, etc.) son algunas de las características que las hacen especialmente interesantes para el aprendizaje de idiomas. En concreto, el análisis de estos registros es de especial interés, pues permite no sólo evaluar diferentes destrezas de los alumnos sino también detectar problemas generales de aprendizaje, identificar determinados perfiles de alumnos, etc. Lamentablemente la gran mayoría de los datos registrados en un mundo virtual no son fácilmente accesibles para el profesor medio de idiomas, pues suelen estar almacenados en ficheros de registro o bases de datos que solo pueden consultarse con lenguajes de programación técnicos. En este trabajo presentamos Virtual World Query Language, un lenguaje específico de dominio que facilita esta labor. Mediante consultas sencillas escritas en un lenguaje simple con una estructura sencilla el profesor podrá acceder tanto a datos generales de los mundos virtuales como a otros datos relacionados con el aprendizaje de idiomas. Nuestra propuesta es que dicho lenguaje pueda usarse para analizar la actividad en estos mundos virtuales mediante la realización de sucesivas consultas que permitan visualizar diferentes aspectos de la interacción que se produjo en dichos mundos virtuales. De esta forma los aspectos de interés detectados en las consultas iniciales más generalistas pueden refinarse con otras más detalladas que permitan contrastar las posibles hipótesis que el profesor realice. Para ello mostramos un ejemplo realizado usando los registros de un mundo virtual OpenSim empleado en la asignatura de Alemán II de la Universidad de Cádiz. En él se evalúan habilidades idiomáticas individuales y colectivas de manera automatizada y atendiendo a criterios objetivos, obteniendo resultados positivos.

PALABRAS CLAVE: lenguaje específico de dominio, indicadores, análisis del aprendizaje, mundos virtuales, aprendizaje de idiomas.

INTRODUCCIÓN

El auge de las Tecnologías de la Información y la Comunicación (TIC) y su asentamiento en los centros de enseñanza, obliga a docentes y a los propios centros a explorar el potencial de diferentes tipos de entornos de aprendizaje virtual (EAV) para utilizarlos con sus estudiantes de idiomas. Las principales ventajas de los EAV consisten en proporcionar a sus usuarios el acceso al aprendizaje de contenidos, servicios y aplicaciones en cualquier momento, en cualquier lugar y a cualquier ritmo. Los EAV más utilizados son los sistemas de gestión de aprendizaje (LMS), los mundos virtuales (VW) y los servicios de red social (SNS). Gracias a estos entornos, la interacción con el idioma ya no se limita al aula, sino que puede llevar fuera de ésta (1).

Además, en los últimos años se ha incrementado el interés por la explotación del potencial educativo de los juegos de ordenador para involucrar a los estudiantes en el aprendizaje de lenguas extranjeras, especialmente fuera del aula (2). Sin embargo, aunque son muchos los investigadores que reconocen el gran potencial educativo y motivacional de los videojuegos, hay pocas experiencias que hayan estudiado directamente su impacto en los estudiantes (3). Esto mismo es aplicable a los VW como Second Life u OpenCobalt (4). El hecho de que la mayoría de los VW no estén basados en software libre hace que para los investigadores sea muy difícil poder analizar las interacciones que allí se producen y analizar su impacto en el proceso de aprendizaje (5 y 6). Sin embargo, tal y como se recoge en (7), esta información es muy valiosa

para ser utilizada en la evaluación y análisis de la interacción de los estudiantes en el VW.

Estas y otras razones nos animaron en primer lugar a diseñar nuestro propio VW basado en OpenSim (servidor 3D de código abierto) y en segundo lugar, a crear un lenguaje específico de dominio (DSL) para recuperar los indicadores objetivos para evaluar el aprendizaje de lenguas extranjeras que tiene lugar en el VW.

Con este propósito desarrollamos VWQL (Virtual World Query Language) y EvalSim. VWQL es un lenguaje para diseñar análisis del aprendizaje que se lleva a cabo en un VW, mientras que EvalSim es una herramienta que acepta las consultas escritas en VWQL. De esta manera los docentes podrán fácilmente obtener indicadores a partir de la información almacenada en la base de datos sin requerirle habilidades técnicas en bases de datos o computación.

El resto del artículo se organiza de la siguiente manera: en la sección dos se explica la metodología y la configuración del experimento; en la sección tres se presenta el estudio de caso; y en la quinta y última sección las conclusiones.

METODOLOGÍA Y CONFIGURACIÓN DEL EXPERIMENTO

El VW que se ha diseñado para este estudio de caso se basa en un videojuego 3D llamado Saturn, en el que los estudiantes tenían que realizar como tarea una compra conjunta. Mientras que el jugador A desempeña el papel de cliente, el jugador B desempeña el papel de dependiente de la

tienda. Durante la partida ambos jugadores deben coordinar sus acciones comunicándose a través del chat de texto. En la figura 1 se puede observar a uno de los estudiantes desempeñando el papel de cliente (en el lado izquierdo) y a otro estudiante desempeñando el papel de dependiente (en el lado derecho).



Figura 1: Dos estudiante jugando una partida en Saturn

Antes de explicar el estudio de caso es necesario presentar VWQL. VWQL es el DSL que se ha desarrollado para recuperar indicadores objetivos del VW basado en OpenSim y que se ha utilizado en el estudio de caso. Además se ha desarrollado EvalSim, un sistema para procesar las consultas escritas en VWQL. Para su desarrollo se ha seguido un enfoque dirigido por modelos (8), como base para modelar procedimientos de obtención de los indicadores que se requieran. EvalSim ha sido creado usando Xtext (7) dentro del Eclipse Modeling Framework. EvalSim está disponible como software de código abierto bajo los términos de la Licencia Pública General de GNU.

La sintaxis de VWQL se muestra en la consulta 1. La primera línea indica el nombre que se quiere dar a la evidencia; la segunda línea indica de qué estudiantes (uno, varios o todos) se quiere obtener la información; y la tercera línea indica qué indicador queremos obtener (palabras, frases, palabras individuales, turnos, tiempo jugado y puntos).

```
Evidence name of the evidence:
get students [id of the student]
show (words [dict] | sentences | single | turns | time | points)
```

Consulta 1: Sintaxis de VWQL versión 0.1

ESTUDIO DE CASO

El estudio de caso que aquí se analiza se llevó a cabo en un instituto, en concreto en un curso obligatorio de alemán del Marco Común Europeo de Referencia de nivel B1 (MCER). El experimento se desarrolló con 5 estudiantes que jugaron varios partidos en el VW. A continuación vamos a describir los dos ejemplos de consultas de información que se lanzaron durante la experiencia.

Con el fin de detectar la capacidad de los estudiantes para comunicarse en un idioma extranjero, se estableció la siguiente hipótesis: *un estudiante tuvo dificultad para hacerse entender si necesitó dos o más frases, por turno, para comunicarse con su compañero de equipo.*

En línea con esta hipótesis, consideramos el ritmo hablando como un indicador interesante, comparando la cantidad de tiempo que cada estudiante jugó con la cantidad de frases que escribió. Para obtener este indicador hemos diseñado este primer fórmula (consulta 2).

```
Evidence time_sentences:
get students
show time, sentences
```

Consulta 2: Consulta de tiempo jugado y frases

Tabla 1: Resultados de la consulta del ritmo al hablar

Estudiante	Ritmo (frases por minuto)
Student1	1,22
Student2	1,50
Student3	0,80
Student4	2,22
Student5	1,50

En la tabla 1 se puede observar que casi todos los estudiantes participantes utilizaron para hacerse entender un promedio de 1 frase por minuto. Sólo un estudiante (Student4) escribió 2,22 frases por minuto. Por otra parte, en el gráfico (figura 2), podemos ver que el estudiante escribió muchas más frases en el chat de texto en comparación con el resto de sus compañeros de clase (columna roja). Sin embargo, Student4 no jugó más tiempo que sus compañeros de clase (columna azul). Esto sugiere que probablemente Student4 no tuvo una conversación fluida con su compañero de juegos. Y por lo tanto, probablemente tuvo que volver a escribir y aclarar muchas frases para llevar a cabo con éxito el juego.

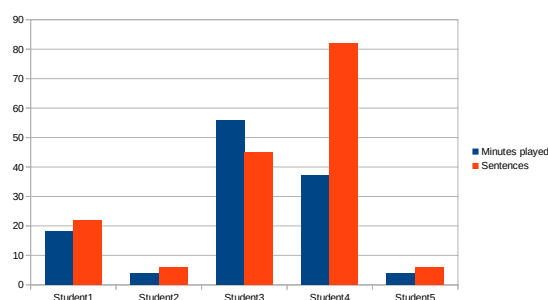


Figura 2: Diagrama de barras con minutos jugados y frases escritas

Pero una diferencia de una frase por minuto no es muy significativa, por lo que para respaldar nuestra evaluación se decidió realizar una nueva consulta considerando en este caso el tamaño de los turnos de cada jugador. Consideramos el tamaño de un turno como el número de frases que tuvo que escribir un estudiante antes de que el otro interviniese. En este caso la consulta utilizada se muestra en la consulta 3.

```
Evidence sentences_turns:
get students
show sentences, turns
```

Consulta 3: Consulta de frases escritas y turnos jugados

El gráfico (figura 3) y la tabla 2 confirman que Student4 necesitó el doble de frases que sus compañeros (por turno) para hacerse entender y llevar a cabo la misma tarea juego. Por lo tanto, esto podría ser un indicador objetivo del hecho de que Student4 tenía mayor dificultad que otros estudiantes

cuando para comunicarse en lengua alemana mediante el chat de texto VW.

Con esta segunda consulta, el supervisor del curso obtuvo datos objetivos para demostrar que Student4 tenía un nivel inferior de comunicación en lengua alemana que sus compañeros de juegos.

Tabla 2: Resultados de la consulta de frases por turno jugado

Estudiante	Media de frases por turno
Student1	1,83
Student2	2,00
Student3	2,14
Student4	4,10
Student5	2,00

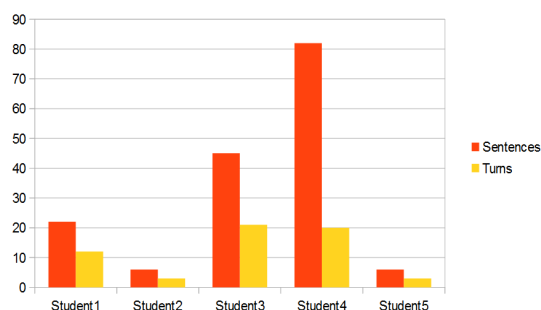


Figura 3: Diagrama de barras con las frases escritas y los turnos jugados

Además, esta relación de frases por turno (obtenida mediante la consulta 3) podría ser un indicador objetivo para evaluar la capacidad de los estudiantes para hacerse entender en el contexto dado (el curso de lengua extranjera alemán - B1).

CONCLUSIONES

Los VW se han vuelto muy populares en el aprendizaje de lenguas extranjeras, ya que apoyan el aprendizaje mediante la participación de estudiantes en juegos atractivos mientras que interactúan en la lengua objetivo de aprendizaje. Además, el docente del curso podría obtener información interesante sobre la interacción de los estudiantes en el VW. Esta información es interesante no sólo en términos de evaluación, sino también para detectar perfiles de alumnos y tendencias generales en el uso de los VW. Por desgracia, esta información generalmente no está a disposición del docente y, aunque lo estuviera, no sería fácil de analizar.

Aquí se propone utilizar VWQL, un DSL para obtener la información de la interacción de los estudiantes con el VW. La sintaxis VWQL es muy simple, por lo que incluso los docentes sin conocimientos avanzados de programación informática podrían obtener fácilmente información muy útil a partir de los registros de VW. Tras escribir consultas en el lenguaje, un informe con los datos solicitados es generado por el sistema de forma automática. De esta manera, se está en condiciones de analizar el comportamiento de los estudiantes en el VW de una manera escalable y de mejorar el aprendizaje.

En el estudio de caso presentado se han descrito las características de la primera versión beta de EvalSim. Los resultados obtenidos son prometedores. A pesar de que es sólo una primera aproximación, la interpretación que hicimos en el estudio de caso de los indicadores, nos ayudó a construir una hipótesis que pudo ser contrastada con una segunda consulta. Por tanto, este estudio ha proporcionado un ejemplo de cómo usando nuestro sistema (EvalSim), el docente puede proponer fácilmente evaluaciones, e ir refinándolas para evaluar el rendimiento individual de los estudiantes en relación con diferentes habilidades, lo que se justifica con cifras objetivas.

Por supuesto, los resultados de este estudio de caso son todavía limitados, y harán falta más estudios para sacar una conclusión más fuerte sobre la validez de la utilización específica de la información recuperada por VWQL.

REFERENCIAS

- Berns, A., Gonzalez-Pardo, A., & Camacho, D. Game-like language learning in 3-D virtual environments. *Computers & Education*. 2013, 60(1), 210-220.
- Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. Model4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns. *Expert Systems with Applications*. 2015, 42(21), 8095 - 8110.
- Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., & Berta, R. Assessment in and of serious games: an overview. *Advances in Human-Computer Interaction*, 2013, 1.
- Fidalgo-Blanco, Á., Sein-Echaluze, M. L., García-Peñalvo, F. J., & Conde, M. Á. Using Learning Analytics to improve teamwork assessment. *Computers in Human Behavior*. 2015, 47, 149-156.
- Brom, C., Preuss, M., & Klement, D. Are educational computer micro-games engaging and effective for knowledge acquisition at high-schools? A quasi-experimental study. *Computers & Education*. 2011, 57(3), 1971-1988.
- Montenegro-Marín, C. E., Cueva-Lovelle, J. M., Sanjuán-Martínez, O., & García-Díaz, V). Domain specific language for the generation of learning management systems modules. *Journal of Web Engineering*. 2012, 11(1), 23-50.
- Eysholdt, M., & Behrens, H. Xtext: implement your language faster than the quick and dirty way. In *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*. 2010, 307-309.
- Schmidt, D. C. (2006). Model-driven engineering. *Computer-IEEE Computer Society*. 2006, 39(2), 25.

AGRADECIMIENTOS

Este trabajo ha sido financiado por la Unión Europea a través de los proyectos OpenDiscoverySpace (CIP-ICT-PSP-2011-5) y UBI CAMP (526843 LLP-1-2012 Es-Erasmus-ESMO).

Bibliografía

- Palomo-Duarte, M., Berns, A., Dodero, J.M., and Ruiz-Rube, I., “Análisis de interacción en entornos virtuales orientados al aprendizaje de idiomas,” *CINAIC 2015, Proceedings of III Congreso Internacional sobre aprendizaje, innovación y competitividad. Madrid, Spain, 14-16 October 2015*, vol. 1, pp. 597–601, 2015.
- Molka-Danielsen, J. and Deutschmann, M., *Learning and teaching in the virtual world of Second Life*. Tapir Academic Press, 2009.
- P. Svensson, “Virtual worlds as arenas for language learning,” *Article in Language learning online: Towards best practice, January 2003*, vol. 10, no. 1, pp. 123–142, 2003.
- Berns, A., Gonzalez-Pardo, A., and Camacho, D., “Game-like language learning in 3-d virtual environments,” *Article in Computers & Education, January 2013*, vol. 60, no. 1, pp. 210–220, 2013.
- , “Implementing the use of virtual worlds in the teaching of foreign languages (level a1),” *EUROCALL 2012, Proceedings of Learning a Language in Virtual Worlds: A Review of Innovation and ICT in Language Teaching Methodology, Warsaw, Poland, 17 November 2011*, pp. 33–40, 2011.
- , “Designing serious games for foreign language learning,” *Proceedings of 4th International Conference ICT for Language Learning, Florence, Italy, 20-21 October 2011*, 2011.
- Palomo-Duarte, M., Berns, A., Yañez Escolano, A., and Dodero, J.M., “Identifying writing profiles in game-based language learning using data mining,” *TEEM’15, Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 7-9 October 2015*, pp. 263–270, 2015.
- Berns, A., Palomo-Duarte, M., Dodero, J.M., and Valero-Franco, C., “Using a 3d online game to assess students’ foreign language acquisition and communicative competence,” *EC-TEL 2013, Proceedings of the 8th European Conference on Scaling Up Learning for Sustained Impact, on Technology Enhanced Learning, Paphos, Cyprus, 17-21 September 2013*, vol. 8095, pp. 19–31, 2013.
- Berns, A., Rodriguez, F., and Gomez, R., “Collaborative learning in 3-d virtual environments,” *WorldCALL 2013, Proceedings of Global perspectives on computer-assisted language learning. Glasgow, United Kingdom, 10-13 July 2013*, pp. 17–20, 2013.
- Berns, A. and Palomo-Duarte, M., “Collaborative learning and foreign language acquisition through 3-d game-like applications,” *International Conference ICT for Language Learning: Conference Proceedings. 5th Conference Edition. Florence, Italy, 15-16 November 2012*, 2012.
- Balderas, A., Berns, A., Palomo-Duarte, M., Dodero, J.M., Gómez-Sánchez, R., and Ruiz-Rube, I., “A domain specific language to retrieve objective indicators for foreign language learning in virtual worlds,” *TEEM’15, Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 7-9 October 2015*, pp. 675–680, 2015.

- Balderas, A., Galán-Piñero, Á., Caballero-Hernández, J. A., Gómez-Rodríguez, G., Dodero, J. M., and Palomo-Duarte, M., “Domain-driven competence assessment in virtual learning environments. application to planning and time management skills,” *SIIE 2014, Proceedings of XVI Simposio Internacional de Informática Educativa, La Rioja, Spain, 12-14 November 2014*, pp. 121–126, 2014.
- CCOO, “Tablas salariales IV Convenio Colectivo PAS UCA,” 2010, <http://www.uca.es/sindicato/ccoo/documentos/tabla-salarial-pas-laboral-2010.pdf>, accessed 2017-08-10.
- L. Bettini, *Implementing domain-specific languages with Xtext and Xtend*. Packt Publishing Ltd, 2016.
- Eclipse Modeling Framework, EMF, “Página oficial de Eclipse Modeling Framework,” <http://www.eclipse.org/modeling/emf/>, accessed 2017-08-10.
- Sirius, “Página oficial de Sirius,” <http://www.eclipse.org/sirius/>, accessed 2017-08-10.
- Git, “Página oficial de Git,” <https://git-scm.com/>, accessed 2017-08-10.
- EGit, “Página oficial de EGit, Eclipse,” <http://www.eclipse.org/egit/>, accessed 2017-08-10.
- EvalSim, “Repositorio privado de EvalSim,” <https://bitbucket.org/RaulGS/vwql>, accessed 2017-08-10.
- Eclipse Modeling Project, EMP, “Página oficial de Eclipse Modeling Project,” <http://www.eclipse.org/modeling/>, accessed 2017-08-10.
- EcoreTools, “Página oficial de EcoreTools,” <http://www.eclipse.org/ecoretools/>, accessed 2017-08-10.
- Xtext, “Página oficial de Xtext,” <http://www.eclipse.org/Xtext/>, accessed 2017-08-10.
- Xtend, “Página oficial de Xtend,” <http://www.eclipse.org/xtend/>, accessed 2017-08-10.
- Pentaho Data Integration, Kettle, “Página oficial de Pentaho,” <http://community.pentaho.com/projects/data-integration/>, accessed 2017-08-10.
- MySQL, “Página oficial de MySQL,” <https://www.mysql.com/>, accessed 2017-08-10.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”). To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.